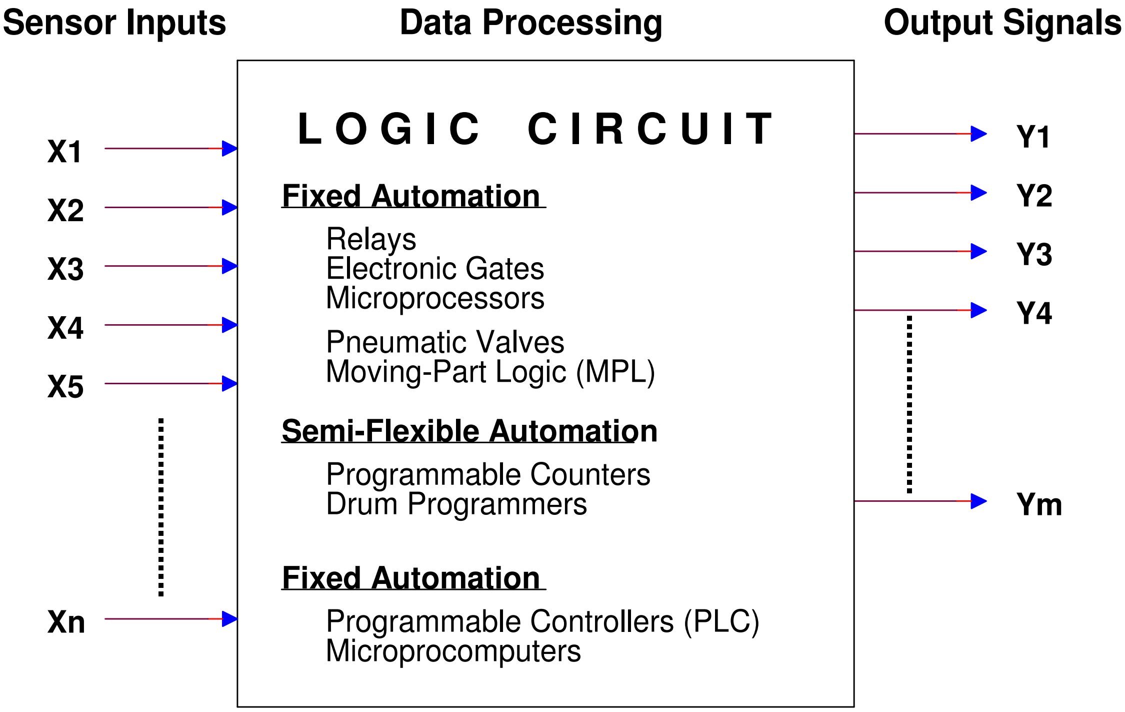
INDUSTRIAL AUTOMATION

SEQUENCE CONTROL



LOGIC CIRCUIT

Fixed Automation

Relays
Electronic Gates
Microprocessors

Pneumatic Valves
Moving-Part Logic (MPL)

Semi-Flexible Automation

Programmable Counters
Drum Programmers

Fixed Automation

Programmable Controllers (PLC)
Microprocomputers

TYPICAL OUTPUT DEVICES

Pneumatic or Hydraulic Cylinders
Pneumatic or Hydraulic Motors
Solenoid Valves
Electric Motors

- Pumps
- Conveyor Belts
- Lifting Devices
- Robot Arms

Heaters
Timers
Lamps
Audioble Signals (Bells, Buzzers, Sirens)
Numerical Displays

הטכניון - מכון טכנולוגי לישראל **הפקולטה להנדסת מכונות**
אוקטובר 2005

אוטומציה תעשייתית (035008)

דף מידע לסטודנט

מרצה הקורס דן קנושר, טל. 04-8712774 , 0546-484-231
אסיסטנט/ית הקורס

פרק	נושא	מבוא לבקרה	תיעוד קודם
1	אלגברה בוליאנית, פישוט פונקציות בינריות ומימושן, מפות קרנו	14	3
2	בנית דיאגרמת סולם (**Ladder Diagram**) שיטת הקסקדה לממסרים	15	5
3	בקר מתוכנת (**PLC - Programmable Controllers**)	16	10
4	סוגים שונים של אלמנטים בינריים, טכנולוגיות, שימוש למימוש מערכות שערים אלקטרוניים ממסרים (**Relays**) שסתומים פניאומטיים אלמנטים לוגיים פניאומטיים עם חלקים נעים (**MPL**)	14	4
5	שיטת הופמן - מערכות עקיבה בעלות אותות כניסה אקראיים		5,6
6	מערכות פיקוד פניאומטיות שיטת הקסקדה הפניאומטית ניצול שסתומים פניאומטיים כאלמנטים לוגיים שיטות לעצירות חירום של מערכות פניאומטיות (**Emergency STOP**)	13	7
7	תכן מערכות עקיבה מסוג "קלות-לשינוי" (**Hardware Programmers**) מפסק תוף צועד (**Drum Programmers**) מונים מתוכנתים (**Programmable Counters, Step Counters**) טבלת זרימה לבחירת השיטה המועדפת לתכן מערכות עקיבה		9
8	מערכות בינריות שונות קוצבי זמן - טיימרים (**Timers**) , **Trigger Flip-Flops** מונים בינריים (**Binary Counters**) רגיסטר הזזה (**Shift Register**) **Schmitt Triggers** קודים בינריים (**Binary Codes**) , מצפינים (**Encoders**)		8
9	מערכות איטרטיביות (**Iterative Systems**)		8

במדה ויתאפשר, יועברו גם נושאים מתוך הרשימה הבאה (ללא קשר לסדר בו הם רשומים) :

פרק	נושאי השלמה	מבוא לבקרה	תיעוד קודם
10	מפות "קרנו" מרובות משתנים		1
11	מערכות פיקוד פניאומטיות - שיטת "טבלת הזרימה" הפניאומטית	13	6
12	מפעילי תנועה (**Motion Actuators**)	12	10
13	**חיישנים חשמליים ופניאומטיים** (**Sensors**)	12	12
14	**דיאגרמת GRAFCET**		5

חומר עזר

1. ספר : **Industrial Automation, by D. Pessen, John Wiley & Sons, 1989**
2. ספר : מבוא לבקרה ואוטומציה (כרקע קדם והשלמה חלקית של מספר נושאים), הוצאת "מכלול"
3. אוסף שקפי ההרצאות : תקליטור שמופץ לסטודנטים בתחילת הסימסטר
3. תרגילי בית להגשה : תקליטור שמופץ לסטודנטים בתחילת הסימסטר (אותו תקליטור)
4. כמו כן, אפשר להיעזר בספרים שונים על תורת המיתוג (לפי מספר קטלוגי 621.3.06 בספרית הפקולטה).

הערה : מומלץ לא להיעזר בתקליטור לא עדכני או בחוברות שנמכרו בסימסטרים קודמים, מאחר וחומר העזר ותרגילי הבית מתעדכן כמעט כל שנה.

סיוע מחוץ לשעות ההרצאה

בתחילת הסימסטר יימסרו שעות הקבלה של אסיסטנט/ית הקורס. במידת האפשר הן יותאמו לזמנים הנוחים יותר לסטודנטים. ניתן גם לפנות לייעוץ וקבלת מידע אל המרצה. פרטים מדויקים יימסרו בתחילת הסימסטר.

תרגילי בית

הגשת תרגילי הבית היא **חובה.**
רשימת התרגילים להגשה תימסר בסיום כל הרצאה. התרגילים יוגשו - בזוגות **קבועים** - שבוע ימים לאחר מכן, בתא המיועד לכך (מסומן "אוטומציה תעשייתית"). במקרים של מניעות חמורות מוצדקות (שירות מילואים, מחלה וכו'), יש להגיש לאסיסטנט/ית אישור מתאים. פרט למקרים אלה – הציון לגבי תרגיל שלא הוגש יהיה 0.
תרגילי הבית ייבדקו בידי האסיסטנט/ית. בכל תרגיל יסומנו השגיאות (במדה ויהיו) וציונו. התרגילים יוחזרו בדר"כ שבוע ימים לאחר הגשתם. בכל מקרה של אי בהירות בנושא התרגילים, יש לפנות אל האסיסטנט/ית בלבד.

ציון תרגילי בית

הציון הממוצע של תרגילי הבית ייקבע על סמך (**n-2**) מתוך **n** התרגילים שיינתנו בסימסטר, עם מתן עדיפות לתרגילים בעלי הציונים הגבוהים ביותר. הגשת פחות מ-(**n-2**) מהתרגילים תפגע בהכרח בציון.
במקרים בהם יילקח בחשבון, יהיה לציון התרגילים משקל של 15% בחישוב הציון הסופי (ראה הסבר בסעיף "ציון סופי" למטה).

בחינת הסימסטר

הבחינה הסופית תתקיים ביום ______________, (מועד א'), וביום ____________________ (מועד ב').
בבחינה ניתן להיעזר בחומר פתוח אישי.

ציון סופי

ציון תרגילי הבית לא יילקח בחשבון במקרים הבאים :

- תרגילי הבית לא הוגשו עקב מניעות חמורות.
- ציון תרגילי הבית גבוה מציון בחינת הסימסטר בלמעלה מ-25% .

במקרים הללו יהיה לציון הבחינה משקל 100% מהציון הסופי.
במקרים בהם ציון תרגילי הבית יילקח בחשבון (15%), יהיה לציון הבחינה משקל 85% מהציון הסופי.

אתר התיעוד באינטרנט – הנדסת מכונות

Meeng.technion.ac.il

↓

studies

↓

here

↓

דוגמה :

רשימת עדכונים ושנויים מגרסה תשס"ד 2003/4

להלן שמות החוברות שעודכנו/נוספו:

	שם החוברת	מחבר	שייך לקורס	מס' קורס	הערות
1	אוטומציה תעשייתית - שקפי הרצאות	קנושר דן	אוטומציה תעשייתית	035008	גרסה מעודכנת
2	אוטומציה תעשייתית – תרגילים	קנושר דן	אוטומציה תעשייתית	035008	גרסה מעודכנת
3	דינמיקה Dynamics	אלטוס אליעזר, מיילס רובין	דינמיקה	034010	גרסה מעודכנת
4	חיכוך ובלאי של חומרים – מאמרים/אנגלית	עציון יצחק	חיכוך שימון ובלאי של חומרים	036031	גרסה מעודכנת
5	מבוא לגיאומטריה תיאורית	בירן אדריאן	שרטוט הנדסי	034004	חוברת חדשה
6	מבוא למכניקת הרצף	מיילס רובין	מבוא למכניקת הרצף	036003	גרסה מעודכנת
7	מכניקת מוצקים 2	ליפשיץ יעקב	מכניקת מוצקים 2	034029	גרסה מעודכנת
8	מערכות הידראוליות ופנוימטיות	אשל ראובן	תכן מערכות הידראוליות 1+2	034206+034205	גרסה מעודכנת
9	תרמואלסטיות יישומית	מיילס רובין	תרמואלסטיות יישומית	035029	גרסה מעודכנת

INTRODUCTION

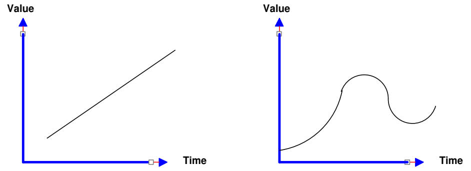
Analog/Digital Differences

Switches and Relays
Pneumatic Switching Valves
Cylinder Control and Sensing
Other Switching Elements

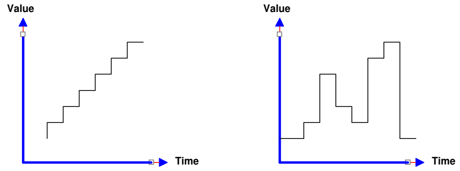
INDUSTRIAL AUTOMATION

Automation of Mechanical Industrial Production Systems
Control Section
On-Off (Binary) Elements and Information
Various Methodes to Obtain Low-Cost Systems

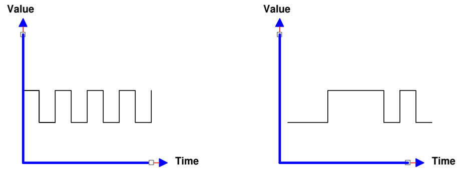
Fig. 0-1 Subject Definition



a. Analog



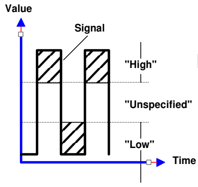
b. Digital



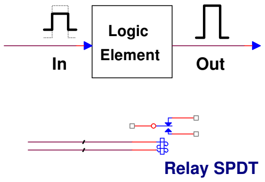
c. Binary (Logic, "Digital")

Fig. 0-2 Information Types

LOGIC IMPLEMENTATION

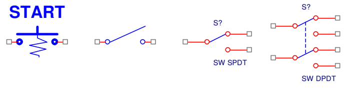


a. Logic Levels

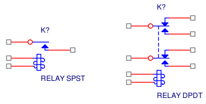


b. Self Correction

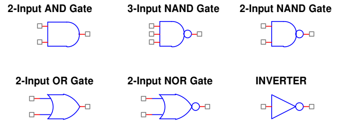
Fig. 0-3 Logic-Levels



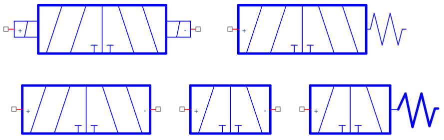
a. Electrical Switches



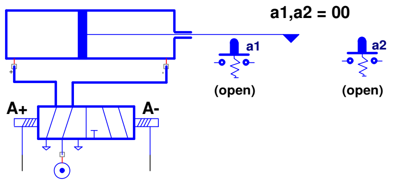
b. Electro-Mechanical Relays



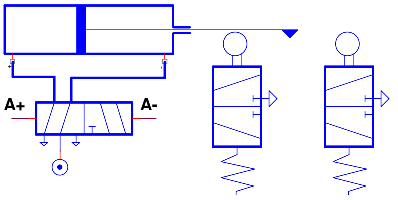
c. Electronic Gates



d. Pneumatic Valves



e. Electrical Limit Switches



f. Pneumatic Limit Valves

BINARY SENSORS :

Flow Switches
Temperature Switches
Level Switches (Height)
Photoelectric Sensors
Pressure Switches
Back-Pressure Sensors
Annular Back-Pressure Sensors

Interruptabel-Jet Sensors
Ultrasonic Sensors
Reed Switches
Threshold Sensors
Magnetic Proximity Sensors
Inductive Proximity Sensors
Inductive Proximity Sensors

g. Various Elements

Fig. 0-4 Binary (Switching) Elements

CHAPTER 1

BOOLEAN ALGEBRA

Binary Basic
Basic Boolean Operations
Truth Table
De-Morgan Functions
Algebric Minimization
Karnaugh Map Minimization

BOOLEAN ALGEBRA

1-1

Logic Theory Implementation in Switching
George Boole
False and True replaced by 0 and 1
Binary Items and Variables
Logic , Boolean , Binary , Digital
Switch or Relay ON/OFF Position :
- **Contacts : Open / Closed**
- **Net : Connected / Not-connected**

Binary sensors (Temperature, Pressure, etc)
Representations :
- **"0 / "1"**
- **"ON" / "OFF"**
- **"HIGH" / "LOW"**

a. Basic Definitions

AND	OR	NOT
0.0 = 0	0+0 = 0	0' = 1
0.1 = 0	0+1 = 1	1' = 0
1.0 = 0	1+0 = 1	
1.1 = 1	1+1 = 1 {*}	

b. Basic Operators

X.0 = 0	X+0 = X
X.1 = X	X+1 = 1
X.X = X	X+X = X
X.X' = 0	X+X' = 1

X+X.Y = X
X+X'.Y = X+Y
X(Y+Z) = XY+XZ
(X+Y).(X+Z) = X+YZ
XY+X'Z+YZ = XY+X'Z

c. Basic Relations

XY+X'Z+YZ = XY+X'Z

F1 = XY+X'Z+YZ   F2 = XY+X'Z

X	Y	Z	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

d. Minimization - Truth Table

XY+X'Z+YZ = XY+X'Z

F1 = XY+X'Z+YZ   F2 = XY+X'Z

If Z=0 then :

F1 = XY+X'.0+Y.0 = XY   F2 = XY+X'.0 = XY

If Z=1 then :

F1 = XY+X'1+Y.1 =   F2 = XY+X'.1 =
= XY+X'+Y = X'+Y   = X'+XY = X'+Y

e. Minimization - Math Operations

NAND (NOT AND) :	X nand Y = (X.Y)'
NOR (NOT OR) :	X nor Y = (X+Y)'
XOR (Exlussive OR)	X xor Y = X'Y+XY'
INHIBITION	X.Y'
IMPLICATION	X+Y'

f. More Useful Operators

(A.B)' = A'+B'
(A+B)' = A'.B'

Implementations :

(A.B+A'.C)' = (A'+B').(A+C')
(AB(C'+DE'))' = A'+B'+C(D'+E)

g. DeMorgan Theorem

AND , OR , NOT	
AND , NOT	A+B = (A'.B')'
OR , NOT	A.B = (A'+B')'
NAND	A' = (A.A)' = (A NAND A) A+B = (A'.B')' = ((A.A)'.(B.B)')' = = (A NAND A) NAND (B NAND B)
NOR	A' = (A+A)' = (A NOR A) A.B = (A'+B')' = ((A+A)'+(B+B)')' = = (A NOR A) NOR (B NOR B)

h. Universal Systems

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

1. Universal Sum-of-Products :
2. Universal Products of Sum :

3. Minimal Sum-of-Products :
4. Minimal Products of Sum :

1. Universal Sum-of-Products :

f(A,B,C,D) = A'B'CD'+A'B'CD+A'BC'D'+A'BC'D+A'BCD+
+ AB'CD'+AB'CD+ABCD'+ABCD

2. Universal Products of Sum :

f'(A,B,C,D) = A'B'C'D' + A'B'C'D + A'BCD' + AB'C'D' +
+ AB'C'D + ABC'D' + ABC'D

f(A,B,C,D) = (A+B+C+D)(A+B+C+D')(A+B'+C'+D)(A'+B+C+D).
.(A'+B+C+D')(A'+B'+C+D)(A'+B'+C+D')

3. Minimal Sum-of-Products :

f(A,B,C,D) = A'BC'+AC+B'C+CD

4. Minimal Products of Sum :

f(A,B,C,D) = (A'+C)(B+C)(A+B'+C'+D)

i. Functions Basic Representations

Fig. 1-1 : Boolean Algebra Concepts

INSURANCE COMPANY REQUIREMENTS

AN INSURANCE COMPANY SELECTS CLIENTS ACCORDING TO THE FOLLOWING CRITERIA : NATIONALITY, GENDER, AGE AND HAVING DRIVING-LICENSE. CLIENT MUST SATISFY AT LEAST ONE OF THE FOLLOWING CONDITIONS :

ISRAELI
and/or NON-ISRAELI UNDER 50 WITH DRIVING LICENSE
and/or NON-ISRAELI MAIL 50 OR ABOVE
and/or NON-ISRAELI UNDER 50 WITHOUT DRIVING LICENSE

FIND THE MINIMAL REQUIRED CONDITIONS

a. System Definition

CRETERIA	VARIABLE	1	0
NATIONALITY	A	ISRAELI	NON-ISRAELI
GENDER	B	MALE	FEMAIL
AGE	C	50 OR ABOVE	UNDER 50
LICENSE	D	HAS LICENSE	DOESN'T HAVE LICENSE

b. Boolean Variables Assignment

T = A+A'C'D+A'BC+A'C'D' = A+A'(C'D+BC+C'D')
= A+C'D+BC+C'D'= A+C'(D'+D)+BC= A+C'+CB= A+B+C'

c. Direct Minimzation

T = A+A'B'C'D'+A'B'C'D+A'BC'D'+A'BC'D+A'BCD'+A'BCD
= A+$\cancel{A'}$(B'C'D'+B'C'D+BC'D'+BC'D+BCD'+BCD)' =
= A+B'C'D'+B'C'D+BC'D'+BC'D+BCD'+BCD' =
= A+B'C'(D'+D)+BC'(D'+D)+BC(D'+D)'
= A+B'C'+B.C'+BC = A + B'C' + B(C'+C) =
= A + B'C' + B
= A + C' + B

e. Truth Table Minimzation

A	B	C	D	T
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

d. Truth Table

SIMPLIFIED RESULT :

ISRAELI
and/or UNDER 50
and/or MALE

f. Minimal Requirements

Fig. 1-2 : Algebric Minimization (1)

AN OCTAL NUMBER IS DEFINED BY THE BINARY COMBINATION X2,X1,X0

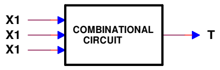
WHERE N(X2,X1,X0)=4.X2+2.X1+1.X0

A COMBINATIONAL SYSTEM DETECETS IF THE NUMBER IS DIVIDABLE BY EITHER 2 OR 3

a. System Definition

X2	X1	X0	T
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

c. Truth-Table



b. Block Diagram

T = X2'X1'X0'+X2'X1.X0'+X2'X1.X0 +
+ X2.X1'X0'+X2.X1.X0'=
=X0'(X2'X1'+X2'X1+X2X1'+X2X1)+X2'X1X0=
=X0'(X2'(X1'+X1)+X2(X1'+X1))+X2'X1X0=
=X0'(X2'+X2)+X2'X1X0=
=X0'+X2'X1X0=
=X0'+X2'X1

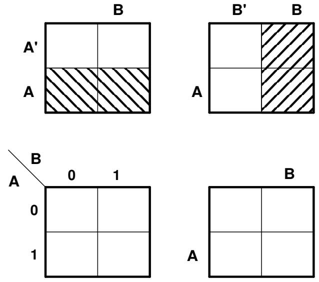
d. Minimization Steps

Fig. 1-3 : Algebric Minimization (2)

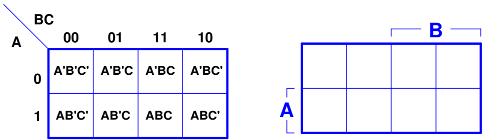
Karnaugh Maps

- Visual method for boolean functions minimization
- Each variable occupies half map
- n variables split map into 2^n basic cells, named map Elements
- Each element is represented by a n-variable boolean product, named "Minterm"
- Any two adjacent elements are represented by two "Adjacent" minterms
- Merging two adjacent elements produces a 2-elements cell, represented by a product of (n-1) variables
- Two adjacent 2-element cells may be merged Into a 4-elements cell, represented by a product of n-2 variables, and so on

a. Map Concepts and Basic Properties

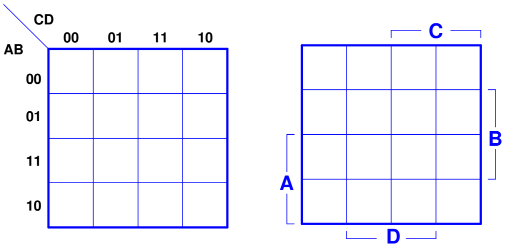


b. 2-Variables Map Representations



	00	01	11	10
0	A'B'C'	A'B'C	A'BC	A'BC'
1	AB'C'	AB'C	ABC	ABC'

c. 3-Variables Map Representations



d. 4-Variables Map Representations

Fig. 1-4 : Karnaugh Maps Concept

GROUPS SELECTION - ESSENTIAL AND NON-ESSENTIAL CANDIDATES

An international company has to hire employees that will be available, as a group, to support the following 6 languages :

Hebrew , English , Russian , Arabic , Rumanian , French

Of course, the company target is to hire the minimal number of employees, and get the required support.

5 candidates look for that job. Next table specifies the language knowledge of each candidate :

Candidate A	Candidate B	Candidate C	Candidate D	Candidate E
Hebrew English Russian Arabic	Hebrew English Romanian	Russian Arabic	Hebrew French	Arabic Russian

There are many available combibations, but we can make the selection easier :

1. Check if there are ESSENTIAL candidates, that is to say candidate that support at least a single language, that none of the others does. These candidates are called ESSENTIAL, since are non-replacable by any of the other candidates.

2. Since all ESSENTAIL candidates (if there are any) must be a part of any selection option, we first identify them and select them.

3. NON_ESSENTIAL candidates doesn't mean that they will not be selected. NON_ESSNTIAL. Since any of them may be replcaed by others, no one can decide - on first glance - which of them must be selected,or not.

4. Actually, some (or all) NON_ESSENTIAL will have to be selected in case that the ESSENTIAL candidates do support all required languages.

5. We can see easily that French is suppoter by candidate D only, and Romanian is supported by candidate B only. This maked D and B ESSNTIAL. Each of all other languages is supported by al least 2 candidates, so all those candidates are NON_ESSENTIAL.TIAL.

6. Selection of candidates B and D (ESSENTIAL) cover more than the 2mentioned languages; they also support Hebrew and English, so we don't have to worry about support of thse 4 langages.

7. Selecting all ESSENTIAL candidates, don't cover the left languages Arabic and Russian. So, we must look for the minimal additional candidates that will completed the.This selection is simpler.

8. In this case, either A or C complete the required support, so the minimal selected group contains 3 candidate (2 ESSENTAIL and 1 ON_ESSENTIAT).

9. There is no other option to the support Arabic and Russian by a single candidate, so there two minimal available selections :

a. { A , B , D}
b. { B , C , D}

10. Principally, both selections are minimal. On the other hand candidate A has higher priority than C since he support 4 labguages and may be ore helpful, and a backup to the other languages that had been covered by the ESSENTIAL candidates.

11. Thsi means that first minimal selction (a) is preferable, which leaves a single minimal solution.

Fig 11-5 : ESSENTIAL and NON_ESSENTIAL concept

1-4

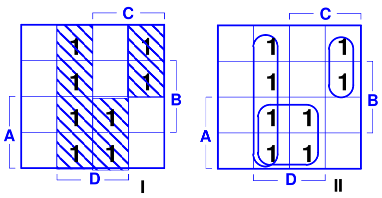


Fig. 1-6 : Example 1

Essential cells : 3
Non-Essential cells : 0
Minimal solutions : 1

F = AD + C'D + A'CD'

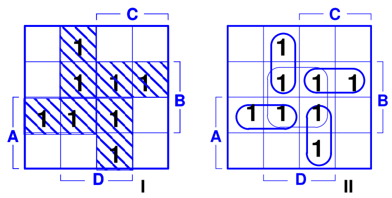


Fig. 1-7 : Example 2

Essential cells : 4
Non-Essential cells : 1
Minimal solutions : 1

F = A'C'D + ACD + A'BC + ABC'

Maps Definitions

- I Function "Area"
- II All (Maximal) Cells
- III Minimal Solution 1
- IV Minimal Solution 2
- V Minimal Solution 4

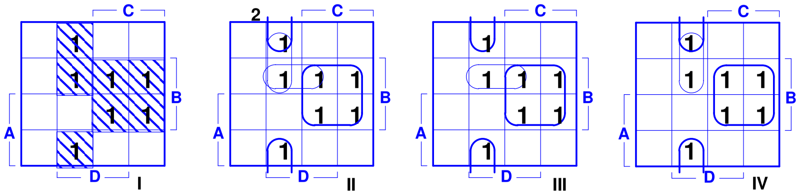


Fig. 1-8: Example 3

Essential cells : 2
Non-Essential cells : 2
Minimal solutions : 2

F1 = BC + B'C'D + A'BD
F2 = BC + B'C'D + A'C'D

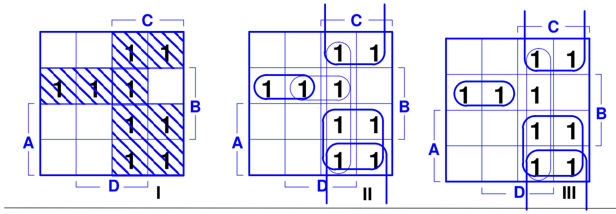


Fig. 1-9 : Example 4

Essential cells : 3
Non-Essential cells : 2
Minimal solutions : 1

F = AC + B'C + A'BC' + CD

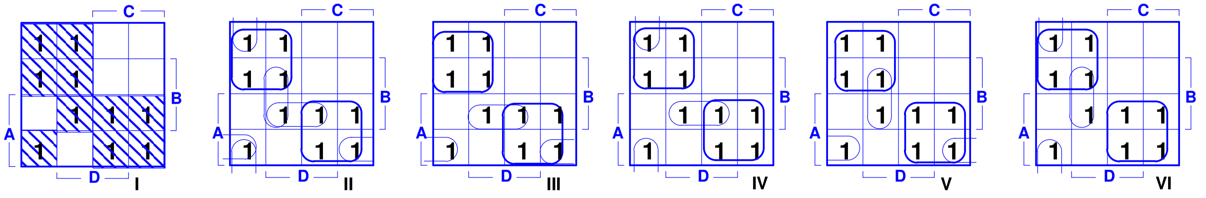


Fig. 1-10 : Example 5

Essential cells : 2
Non-Essential cells : 4
Minimal solutions : 4

F1 = A'C' + AC + ABD + AB'D'
F2 = A'C' + AC + ABD + B'C'D'
F3 = A'C' + AC + BC'D + AB'D'
F4 = A'C' + AC + BC'D + B'C'D'

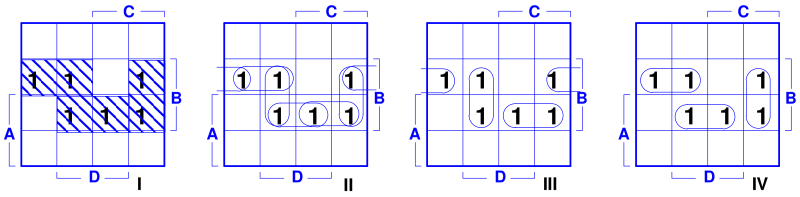


Fig. 1-11 : Example 6

Essential cells : 0
Non-Essential cells : 6
Minimal solutions : 2

F1 = ABC + BC'D + A'BD'
F2 = BCD' + ABD + A'BC'

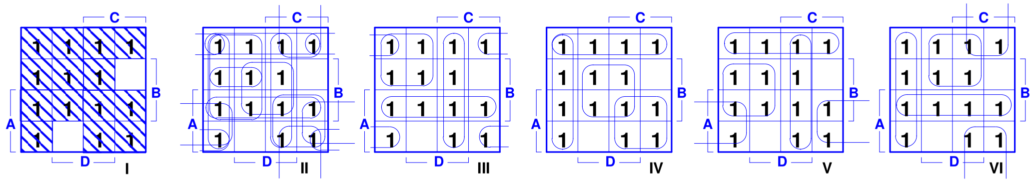


Fig. 1-12 : Example 7

Essential cells : 0
Non-Essential cells : 12
Minimal solutions : 6

F1 = A'C' + B'D'+ AB + CD
F2 = BD+ AC + A'B'+ C'D'
F3 = AD'+ BC'+ CD + A'B'
F4 = A'D + B'C + AB+ C'D'
F5 =
F6 =

Maps Definitions

- I Function "Area"
- II All (Maximal) Cells
- III Minimal Solution 1
- IV Minimal Solution 2
- V Minimal Solution 4

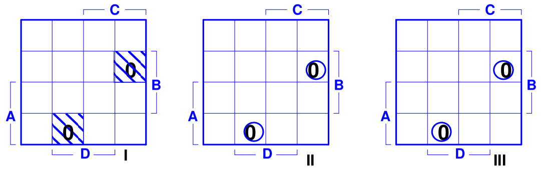


Fig. 1-13 : Example 7.1

Essential cells : 2
Non-Essential cells : 0
Minimal solutions : 1

F' = A'BCD' + AB'C'D

F = (A + B' + C' + D).(A' + B + C + D')

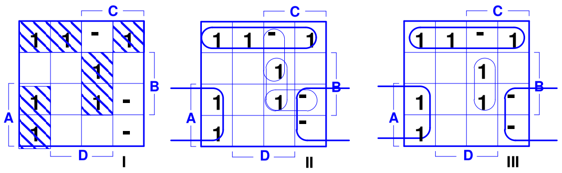


Fig. 1-14 : Example 8

Essential cells : 2
Non-Essential cells : 1 (+2)
Selected Don't Cares : All
Minimal solutions : 1

F = AD' + A'B' + BCD

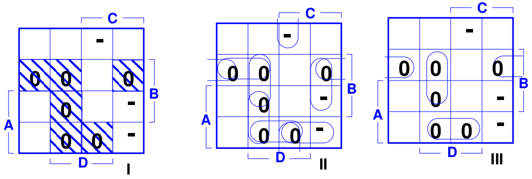


Fig. 1-15 : Example 8.1

Essential cells : 0
Non-Essential cells : 8
Selected Don't Cares : None
Minimal solutions : 1

F' = AB'D + A'BD' + BC'D

F = (A'+B+D').(A+B'+D).(B'+C+D')

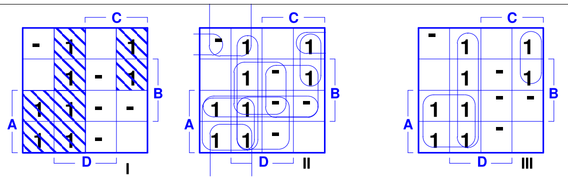


Fig. 1-16: Example 9

Essential cells : 0
Non-Essential cells : 3 (+5)
Selected Don't Cares : None
Minimal solutions : 1

F = AC' + C'D + A'CD'

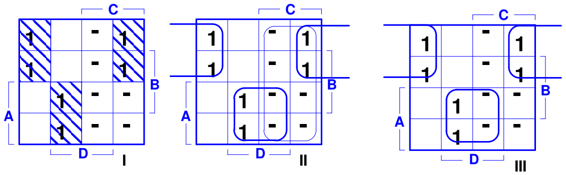


Fig. 1-17 : Example 10

Essential cells : 2
Non-Essential cells : 1
Selected Don't Cares : Partial
Minimal solutions : 1

F = AD + A'D'

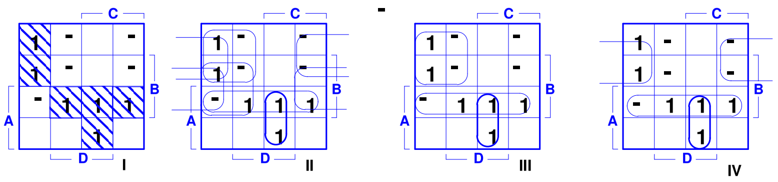


Fig. 1-18 : Example 11

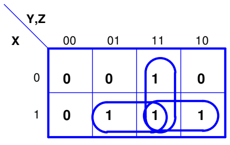
Essential cells : 1
Non-Essential cells : 5
Selected Don't Cares : Partial
Minimal solutions : 2

F1 = ACD + AB + A'C'
F2 = ACD + AB + A'D'
Note : F1 and F2 are equivalent but not identical

1-6

X	Y	Z	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

a. Truth-Table



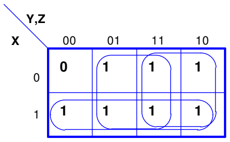
M = XY+XZ+YZ

b. Karnaugh-Map

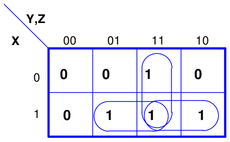
Fig. 1-19 : Majority Function

X	Y	Z	A	B	C
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1

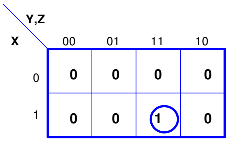
a. Truth-Table



A = X + Y + Z

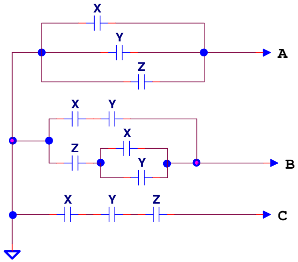


B = XY + XZ + YZ

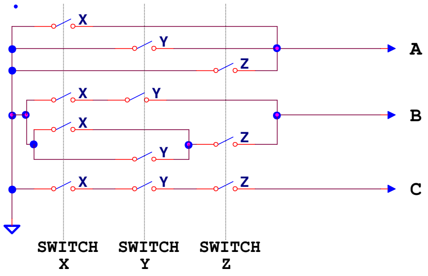


C = XYZ

b. Karnaugh-Maps



c. Contact Circit Realization



d. Electrical Diagram

Fig. 1-20 : Motor Operation Control

(Exer 1-7)

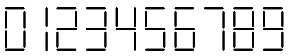
BCD to 7-SEGMENTS CONVERTER

Design a combinational circuit that gets a BCD input, and converts it to 7-Segment display.

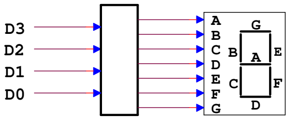
BCD is short form of Binary-Coded-Decimal. It Represents 10 decimal digits (0-9) in 4-bit binary code (0000-1001).

Binary combinations 1010-1111 (above decimal 9) may not appear as inputs, and are refered as don't-care.

a. System Definition



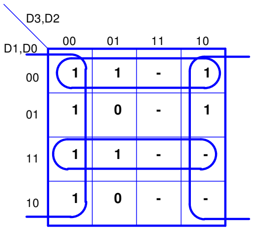
b. 7-Sigments Arrangement



c. Block Diagram

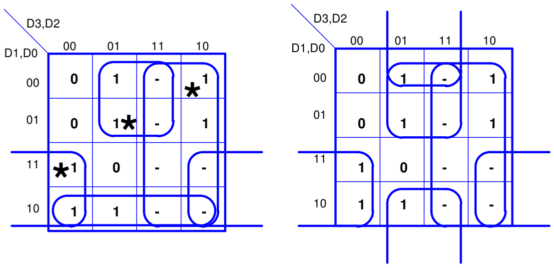
D3	D2	D1	D0	A	B	C	D	E	F	G
0	0	0	0	0				1		
0	0	0	1	0				1		
0	0	1	0	1				1		
0	0	1	1	1				1		
0	1	0	0	1				1		
0	1	0	1	1				0		
0	1	1	0	1				0		
0	1	1	1	0				1		
1	0	0	0	1				1		
1	0	0	1	1				1		
1	0	1	0	-				-		
1	0	1	1	-				-		
1	1	0	0	-				-		
1	1	0	1	-				-		
1	1	1	0	-				-		
1	1	1	1	-				-		

d. Truth-Table (Partial)



E=D2'+D1'D0'+D1.D2

e. Karnaugh Map of "E"



A=D3+D2.D1'+D2'D1+D1.D0'=
=D3+D2.D1'+D2'D1+D2.D0'

f. Karnaugh Map of "A"

Fig. 1-21 : BCD to 7-Segments

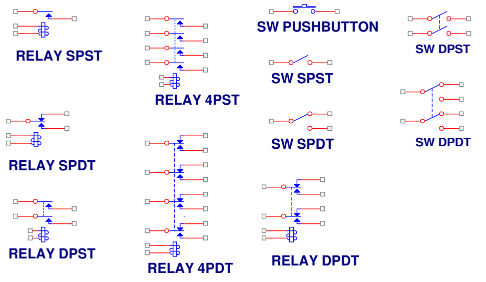
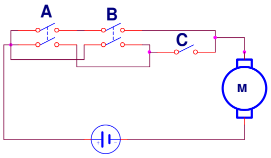


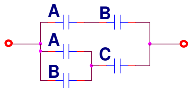
Fig. 1-22 : Various Switches and Relays Types

T = AB + AC + BC = AB + (A + B)C

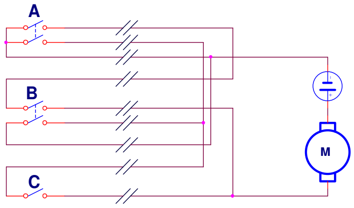
a. Majority Boolean Function



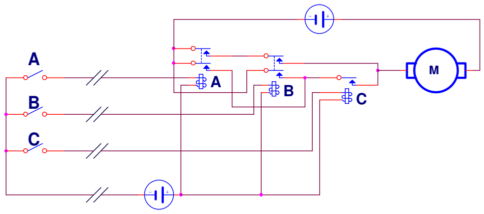
b. Switches-Operated Circuit



c. Contacts Circuit

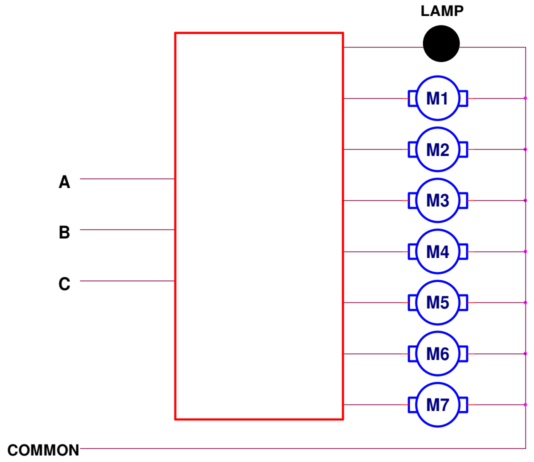


d. Switches Remote Operation
9 Long Lines, Carrying High Current



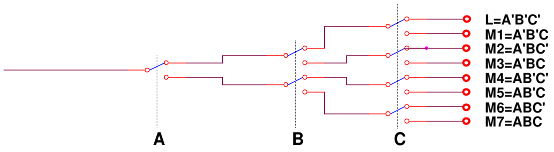
e. Relays Remote Operation
3 Long Lines, Carrying Low Current

Fig. 1-23 : "MAJORITY" Function Circuit

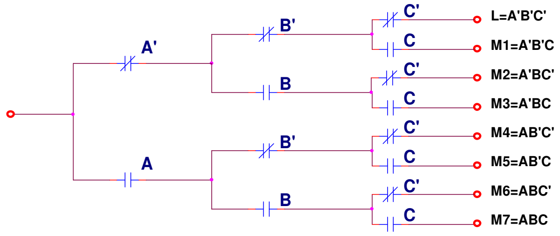


L = A'B'C'
M1 = A'B'C
M2 = A'BC'
M3 = A'BC
M4 = AB'C'
M5 = AB'C
M6 = ABC'
M7 = ABC

a. Block-Diagram

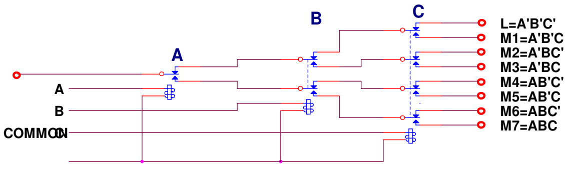


b. Switches-Operated Circuit



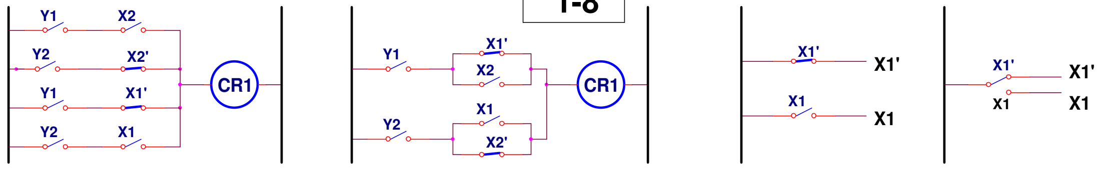
Long lines, high power dissipation

c. Contacts Circuit



d. Relays-Operated Circuit

Fig 1-24 : 3-TO-8 DECODER CIRCUIT



2 Contacts
4 Springs

1 Contact
3 Springs

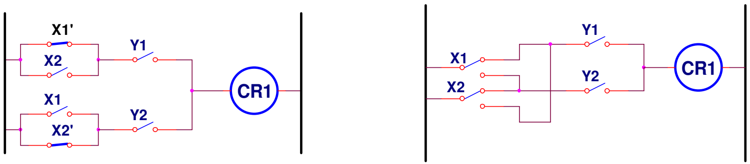
CR1=Y1.X2+Y2.X2'+Y1.X1'+Y2.X1
8 Contacts , 16 Springs

a. Original Function

CR1=Y1(X1'+X2)+Y2(X1+X2')
6 Contacts , 12 Springs

b. Saving Contacts

c. Saving Contact Springs



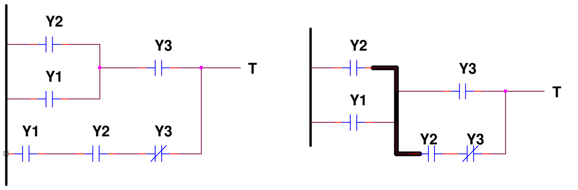
CR1=Y1(X1'+X2)+Y2(X1+X2')
6 Contacts , 12 Springs

d. Changing circuit schema

CR1=Y1(X1'+X2)+Y2(X1+X2')
4 Contacts , 10 Springs

e. Saving Contact Springs

Fig. 1-25 : Saving Contacts and Contacts Springs



T=Y2.Y3+Y1.Y3+Y1.Y2.Y3'

a. Original Function

T=Y2.Y3+Y1.Y3+Y1.Y2.Y3'+(Y2.Y3') = Y2 + Y1.Y3

b. Sneak Path Y2.Y3'

Fig.1-26 : Sneak Path Creation

a. Use of Relay

b. Use of Gates

Fig. 1-27 : Devices Outputs (fan-out)

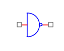
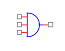
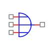
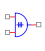
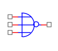
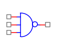
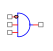
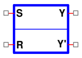
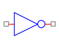
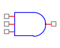
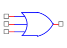
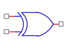
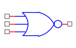
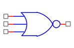
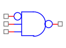
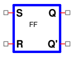
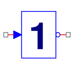
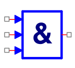
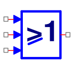
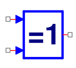
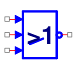
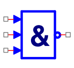
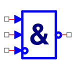
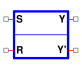
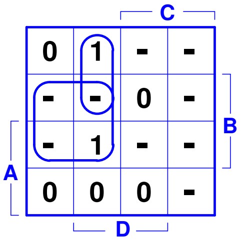
	NOT	AND	OR	EXCLUSIVE OR	NOR	NAND	IMHIBITION	FLIP-FLOP
Old European Symbols								
Old USA Symbols								
New Symbols Recommended by ISO								

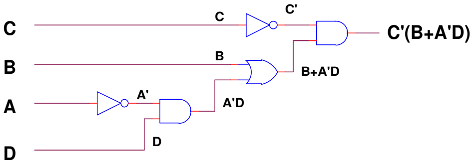
Fig. 1-28 : Logic Gate Symbols



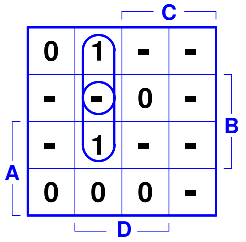
0	1	-	-
-	-	0	-
-	1	-	-
0	0	0	-

T=BC'+A'C'D=
=C'(B+A'D)

a. Prime Cells Selection



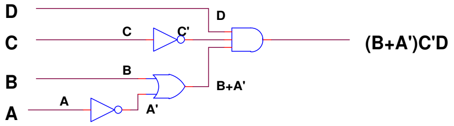
b. Gates Implementation of (a) - 5 Gates



0	1	-	-
-	-	0	-
-	1	-	-
0	0	0	-

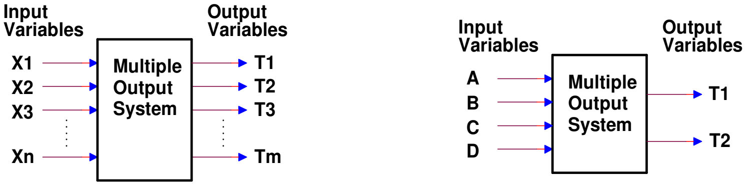
T=BC'D+A'C'D=
=(B+A')C'D

c. Non-Prime Cells Selection



d. Gates Implementation of (c) - 4 Gates

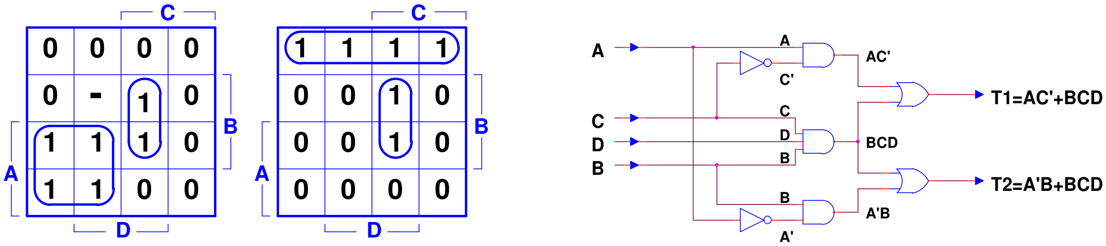
Fig. 1-29 : Minimize by Selecting Non-Prime Cells



a. General Case

b. Specific Case

Fig. 3-30 : Multiple Output System



0	0	0	0
0	-	1	0
1	1	1	0
1	1	0	0

1	1	1	1
0	0	1	0
0	0	1	0
0	0	0	0

T1=AC'+BCD
(Instead of A'C+BD)

T2=A'B+BCD

a. Function T1

b. Function T2

C. Gates Implementation

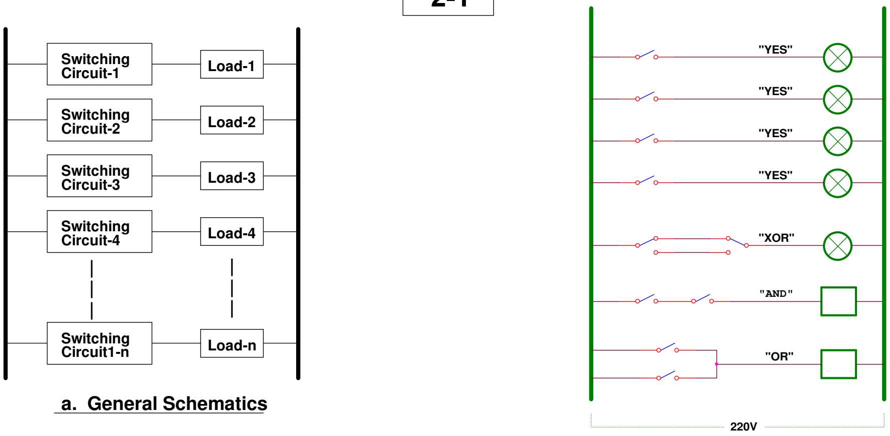
Fig. 1-31 : Minimize by Selecting Non-Prime Cells, in Multiple-Ourtut System

CHAPTER 2

RELAYS CASCADE
SYSTEMS

Ladder Diagram
Relays, Cylinders, Valves Symbols
Sequential Sequences
Relays Cascade Implementation
Huffman Method
Flow Diagram
Primitive and Merged Flow Tables
State Assignment
Output and Excitation Functions

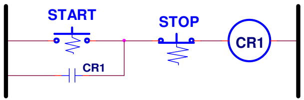
2-1



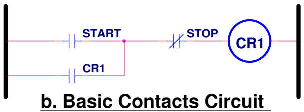
a. General Schematics

b. Example - Home Electric Ladder Network

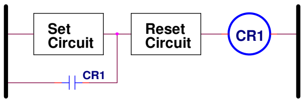
Fig. 2-1 : Ladder Diagram



a. Basic Circuit

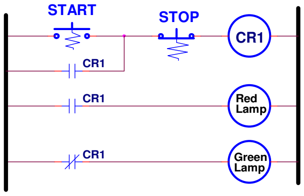


b. Basic Contacts Circuit

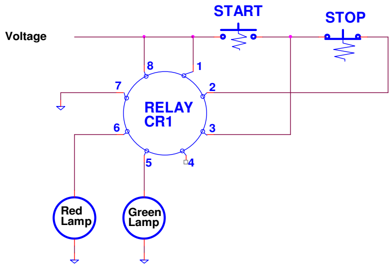


c. General Circuit

Fig. 2-2 : Set-Reset Relay Flip-Flop

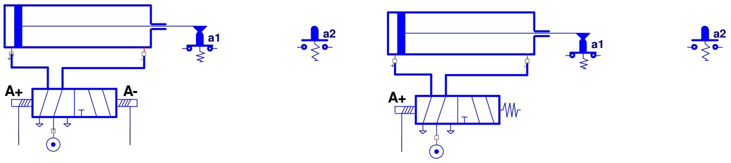


a. Example - Schematic Circuit



b. Example - Wiring Diagram

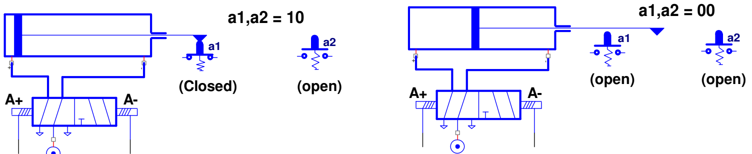
Fig. 2-3 : Flip-Flop Implementation Example



a. Valve Without Return Spring

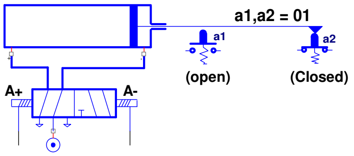
b. Valve With Return Spring

Fig. 2-4 : Cylinder Actuation Valves



a. Position "-"

b. Position "Moving"



c. Position "+"

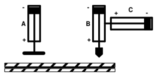
Fig. 2-5 : Positions of Cylinder Limit Switches

a. Normally Open

b. Normally Closed

Fig. 2-6 : Graphic Symbols of Limit Switches

A Sequence of Drilling Two Holes in a Wodden Plate



Cylinder "A" fastens/reases the wooden plate.
Cylinder "B" lowers and lifts a driller..

a. Process Mechanical Representation

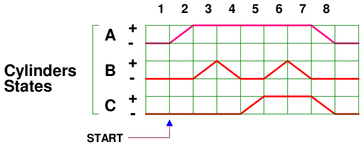
Process starts by pressing START pushbotton

1. Cylinder "A" fastens the wooden plate
2. Cylinder "B" lowers the driller thus performing drilling action
3. Cylinder "B" lists the driller thus disconnecting it from the plate
4. Cylinder "C" shifts the driller to next hole location.
5. Cylinder "B" lowers the driller thus performing drilling action
6. Cylinder "B" lists the driller thus disconnecting it from the plate
5. Cylinder "C" returns the driller to its initial position, and cylinder "A" releases the wooden plate.

b. Sequence Description

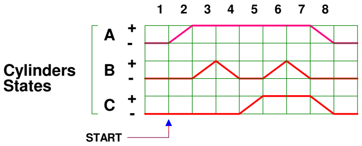
START , A+ ,B+ ,B- , C+ ,B+ ,B- , (A- C-)

c. Sequence Short Representation

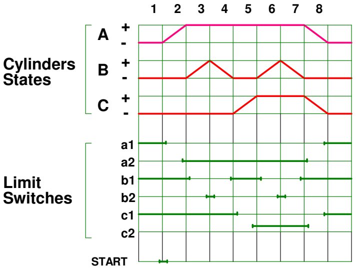


d. Cylinders Sequence Process Chart

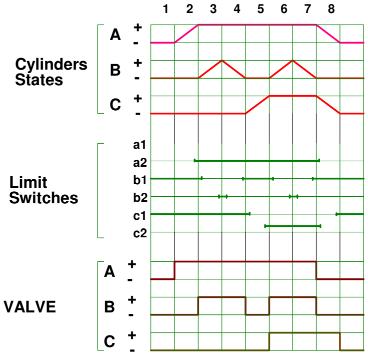
Fig. 2-7 : Process Sequence Representations



A. Cylinders Position Chart



B. Cylinders Position Chart & Contacts



C. Cylinders Position Chart, Contacts & Valves Position

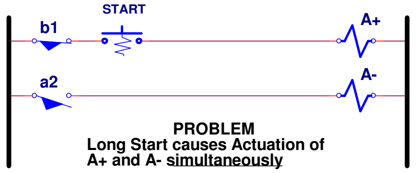
Fig. 2-8 : Sequence Chart Diagram

Fig. 2-2 : "Intuitive" Design Problems

START,A+,A-,B+,10 Sec delay,B-
(STEPS: 1 2 3 4 5)

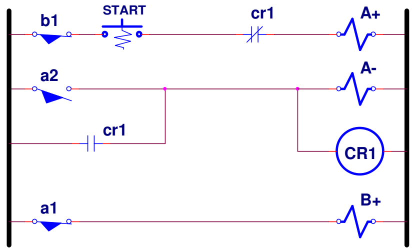
a. Process Sequence (No Return Springs)

b. Step 1 : Actuate A+



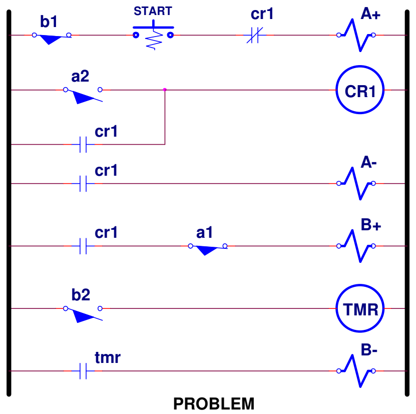
PROBLEM
Long Start causes Actuation of A+ and A- simultaneously

c. Step 2 : Actuate A-



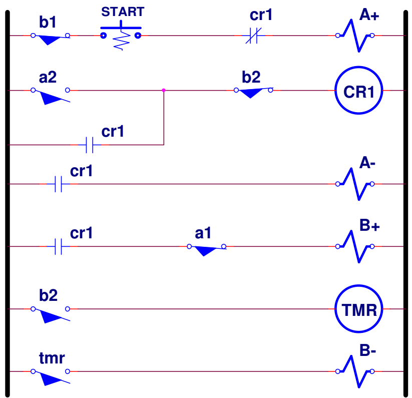
PROBLEM
B+ is also actuated before cycle starts

d. Step 3 : Correct and Actuate B+



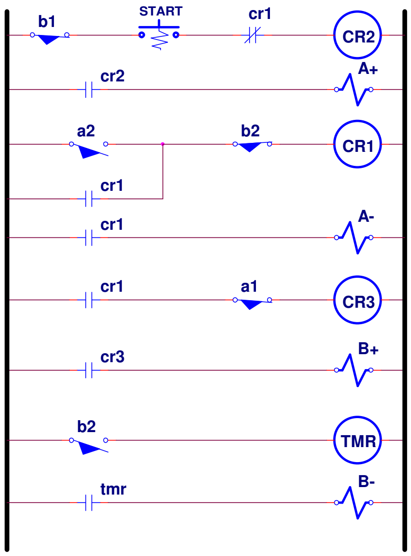
PROBLEM
B+ and B- are actuated simultaneously.
CR1 is actuated infinitily

e. Steps 4-5 : Correct and Actuate TMR and B-



Add reset (b2') to CR1

f. Correct problems of (e)



g. Isolate Limit Swithes from Solenoids Current

Fig. 2-9 : "Intuitive" Design Problems

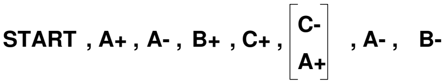
RELAYS CASCADE SYSTEMS

- * Avoid conflicting actuation of a cylinder
- * Distinguish between situations where a cylinder performs more than a single cycle

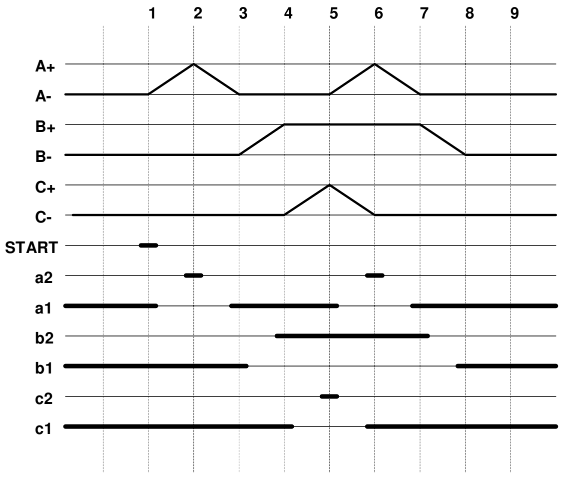
---

- * Maximal size groups (to obtain minimum number of group relays)
- * Same "Letter" may appear no more than once in a group (to avoid conflicted commands to a cylinder)
- * Except for specific cases, groups Partitioning doesn't depend on having or not having valves return springs

Fig. 2-10 : Cascade Method Target & Rules



a. Sequence List Representation



b. Sequence Chart Representation

Fig. 2-11 : Typical Process Definition

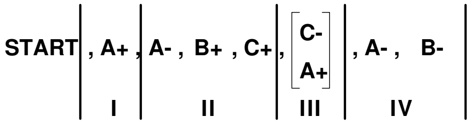


Fig. 2-12 : Typical Groups Partitioning

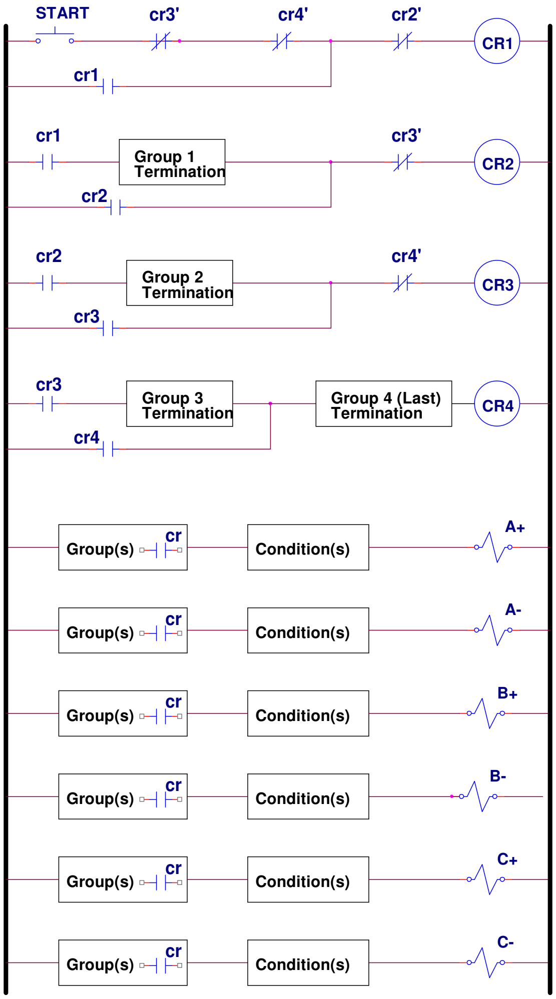


Fig. 2-13 : Typical Initial Cascade Circuit, of 3-Cylinder / 4-Groups Process

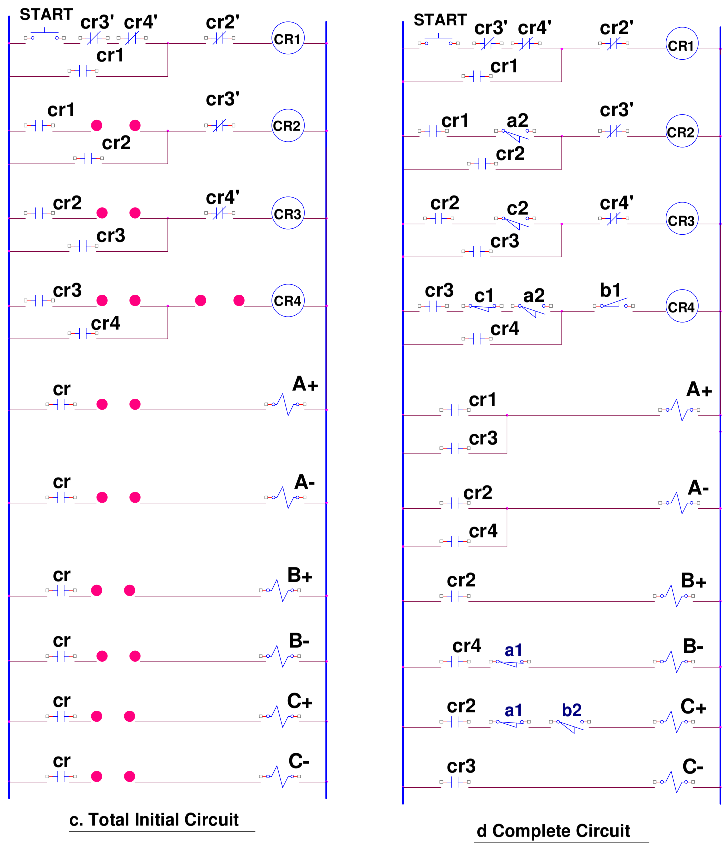
$$\text{START , A+ ,}\begin{bmatrix}A-\\B+\end{bmatrix}\text{, C+ ,}\begin{bmatrix}C-\\A+\end{bmatrix}\text{, A- , B-}$$

1. Cylinders are actuated by 5/2 valves without return springs.
2. Valves are actuated electrically (solenoids).

a. Process Sequence Information

$$\begin{array}{l|l|l|l|l|}\text{START} & \text{A+ ,} & \begin{bmatrix}A-\\B+\end{bmatrix}\text{, C+ ,} & \begin{bmatrix}C-\\A+\end{bmatrix} & \text{, A- , B-}\\ & \text{I} & \text{II} & \text{III} & \text{IV}\end{array}$$

b. Groups Partitions



c. Total Initial Circuit

d Complete Circuit

Fig. 2-14 : Relay Cascade Design Process 1 (No Return Springs)

$$\text{START , A+ ,}\begin{bmatrix}A-\\B+\end{bmatrix}\text{, C+ ,}\begin{bmatrix}C-\\A+\end{bmatrix}\text{, A- , B-}$$

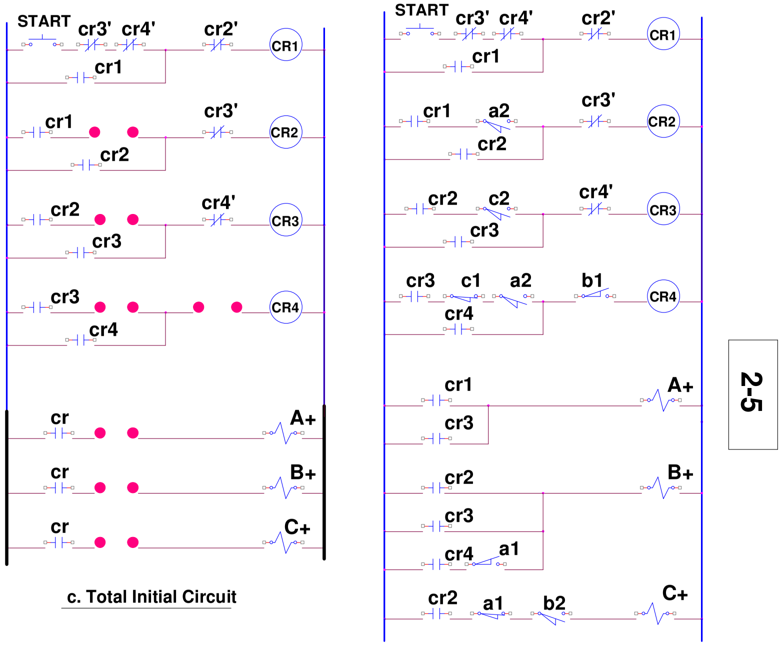
1. Cylinders are actuated by 5/2 valves with return springs.
2. Valves are actuated electrically (solenoids).

a. Process Sequence Information

B+ duration

$$\begin{array}{l|l|l|l|l|}\text{START} & \text{A+ ,} & \begin{bmatrix}A-\\B+\end{bmatrix}\text{, C+ ,} & \begin{bmatrix}C-\\A+\end{bmatrix} & \text{, A- , B-}\\ & \text{I} & \text{II} & \text{III} & \text{IV}\end{array}$$

b. Groups Partitions



c. Total Initial Circuit

d. Complete Circuit

Fig. 2-15 : Relay Cascade Design Process 1 (With Return Springs)

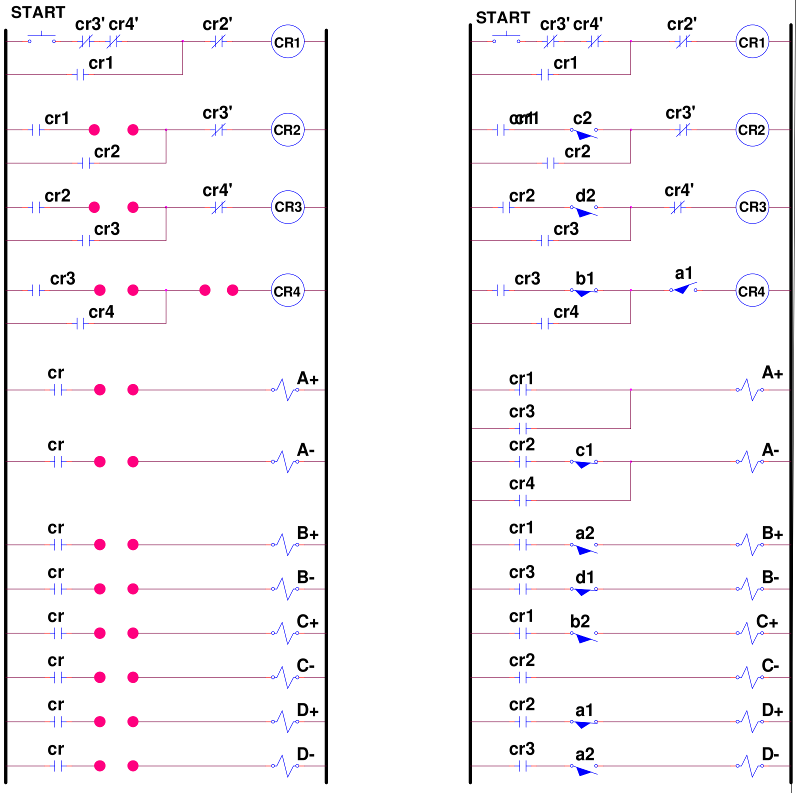
START , A+ , B+ , C+ , C- , A- , D+ , A+ , D- , B- , A- ,

1. Cylinders are actuated by 5/2 valves without return springs.
2. Valves are actuated electrically (solenoids).

a. Process Sequence Information

START	A+ , B+ , C+	C- , A- , D+	A+ , D- , B- ,	A- ,
	I	II	III	(IV)

b. Groups Partition



c. Total Initial Circuit

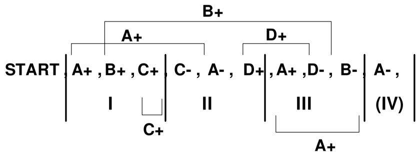
d. Complete Circuit

Fig. 2-16 : Relay Cascade Design Process 2 (Without Return Springs)

START , A+ , B+ , C+ , C- , A- , D+ , A+ , D- , B- , A- ,

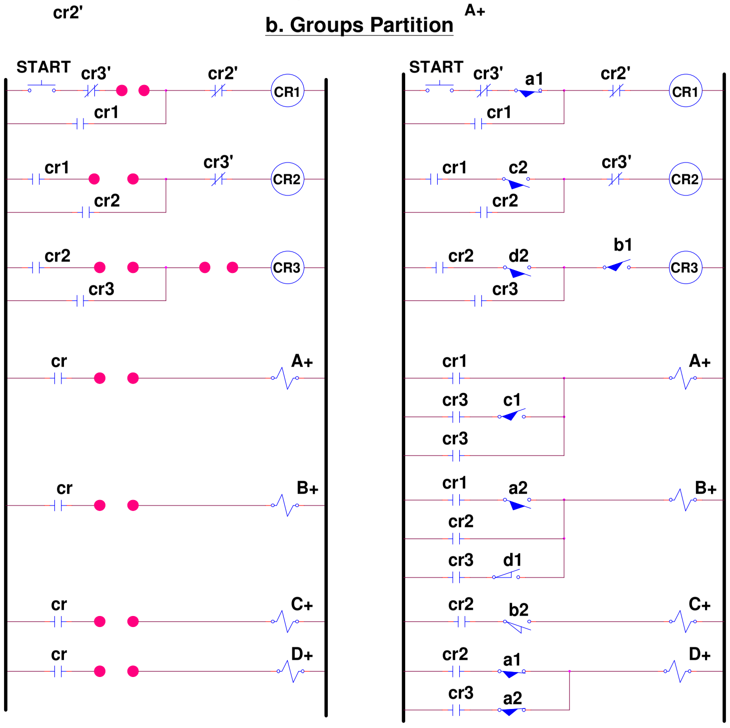
1. Cylinders are actuated by 5/2 valves with return springs.
2. Valves are actuated electrically (solenoids).

a. Process Sequence Information



START	A+ , B+ , C+	C- , A- , D+	, A+ , D- , B-	A- ,
	I	II	III	(IV)

b. Groups Partition



c. Total Initial Circuit

d. Complete Circuit

Notes : CR4 been canceled (useless).
Rset CR3 been corrected
Contact a1 been added to Set CR1

Fig. 2-17 : Relay Cascade Design Process 2 (With Return Springs)

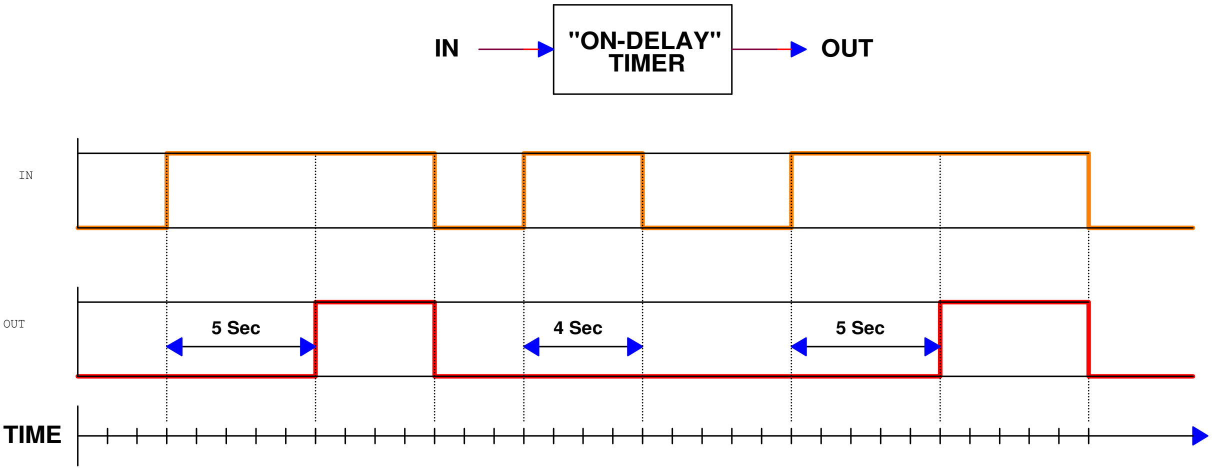
"ON-DELAY" TIMER DEFINITION

As long as input is "0", output is also "0".

When input changes from "0" to "1", output change to "1" is delayed by T sec.

When input changes from "1" to "0", output changes to "0" immediately, and time count resets t

Note : Element behaves as a slow activated relay : it changes to active state only T sec after been energized, but returns to rest state as soon as energizing terminates.



Timing chart of a 5-Sec delay timer (example)

Fig. 2-17 : "On-Delay" Timer Definition

CHAPTER 3

PLC

(Programmable Logic Controller)

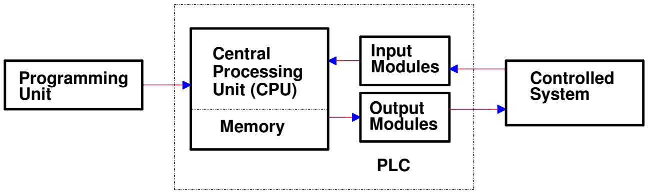
PLC Definition
PLC Hardware Control System
PLC Programming
Programming LIFO Stack
Implementation
Timers and Counters
Modified Programming Commands
Standard I/O

PLC - Programmable Logic Controller

3-1

Micro-computer for controlling industrial processes
First applied at 60th-70th
Implement binary and analog signals, PID
Eliminate need to implement relays and its wiring
Cost economical, requires small space
Contains timer' counters and more
Perfect reliability
Solution to the design of complex systems
Easy to program and easy to change it
Input/output are adapted to high voltage/current devices
Reliable operation under industrial conditions
Built in self diagnostics
May be controlled and monitored by central computer
Not economic for small control processes
Slow but OK for industrial timimg

a. General



b. Block Diagram of PLC

3-1. Programmable Controller (PLC) Representation

Limit Switches
Proximity Switches
Photoelectric Switches
Sensor Switches (Level, Pressure, Temperature etc)
Push-Button Switches
Selector Switches
Relay Contacts

a. Discrete Input Devices to PLC

Contact Relay Coils
Valve Solenoids
Pneumatic or Hydraulic Cylinders and Motors
Lights
Audible Alarms (Bells,Buzzers,Sirens)
Fans
Heaters
Motor Starters

b. Discrete Output Devices Actuated by PLC

12, 24, 48, 120, 230 Volt AC
12, 24, 48, 120, 230 Volt DC
5V DC (TTL Level)
Conract Relay Output

c. Standard Discrete I/O Interface Modules

3-2. PLC Interfaces

Field signal may not be connected directly to PLC
Converts field signals to logic level, and vica versa
Adapted to implement DC and AC signals
Contains protection from voltage transients and noise
Contains protection from oposite polarity connectios
Electrical common isolation
Expensive
Adapted to work in industrial conditions

3-3 Input/Output Modules

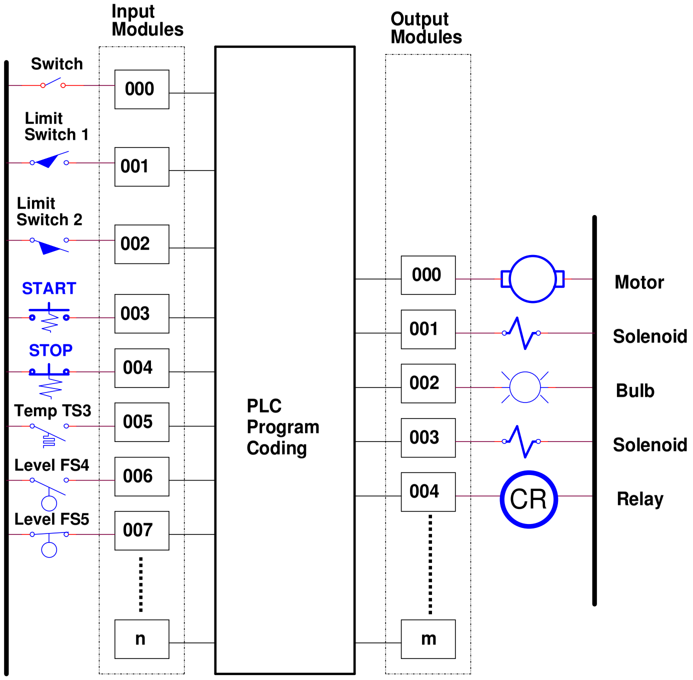
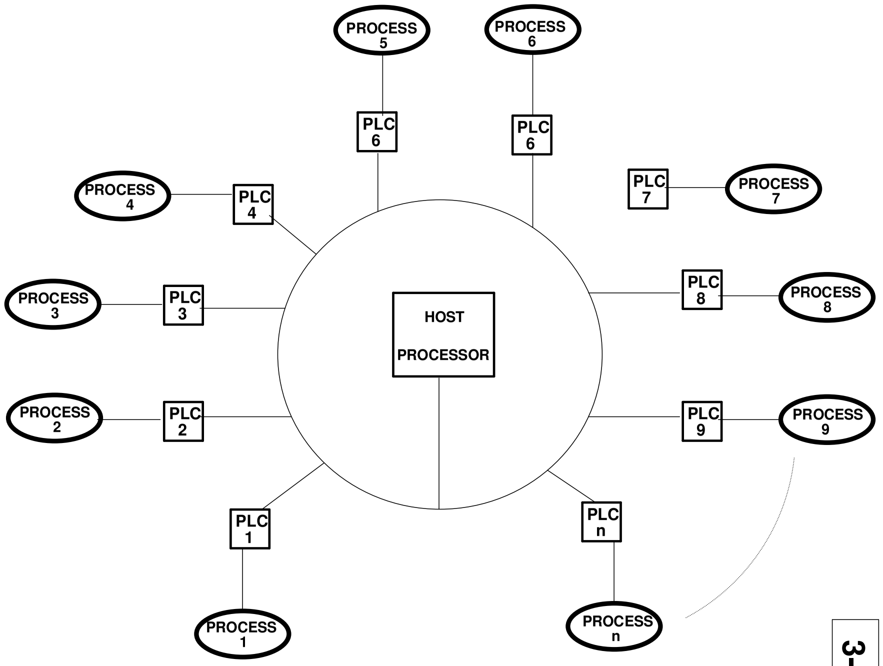
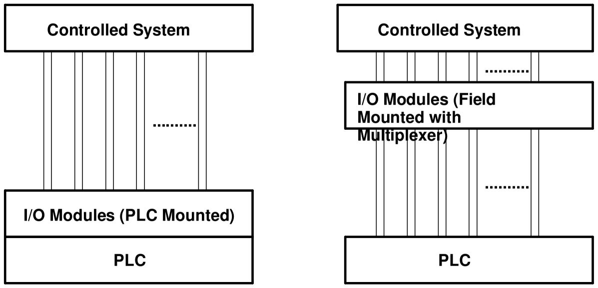


Fig. 3-4 : General I/O Connection Diagram



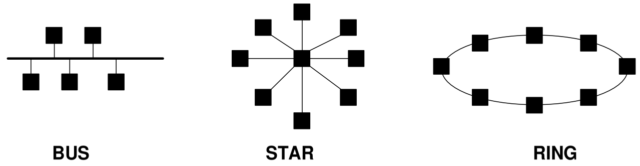
a. PLC Typical Network



a. PLC-Mounted I/O

b. Remote-Mounted I/O

Fig. 3-5 : Different I/O Connection Location



b. Several Possible LAN Topologies

Fig. 3-6: PLC Macro Networks

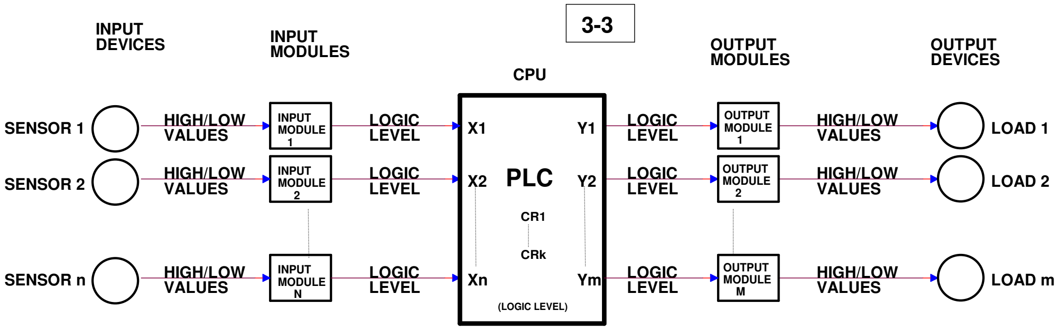


Fig. 3-7 : PLC General Configuration

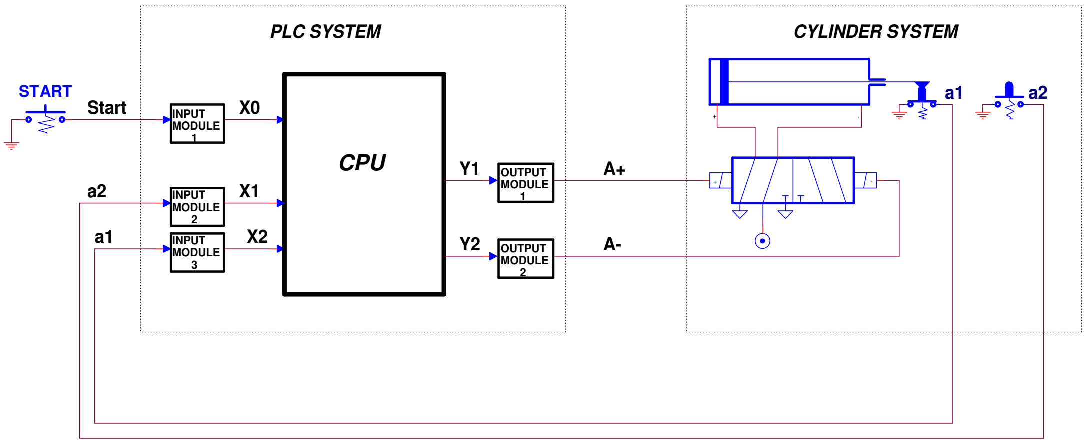


Fig. 3-8 : Typical PLC Control System Example

START , A+ , $\begin{bmatrix} D+ \\ C+ \end{bmatrix}$, C- , A- , B+ , A+ , C+ , B- , D- , $\begin{bmatrix} C- \\ A- \end{bmatrix}$

a. Process sequence

X0	START
X1	a1
X2	a2
X3	b1
X4	b2
X5	c1
X6	c2
X7	d1
X8	d2

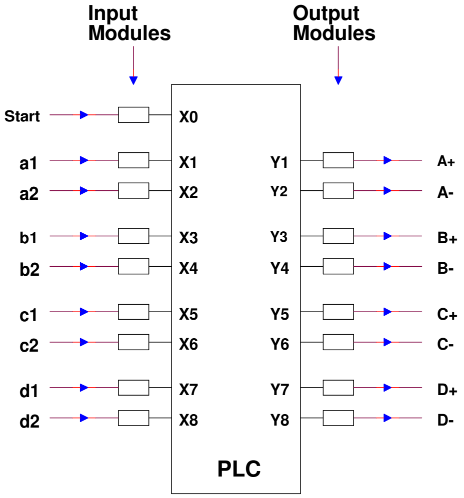
b. Input Table

Y1	A+
Y2	A-
Y3	B+
Y4	B-
Y5	C+
Y6	C-
Y7	D+
Y8	D-

c. Output Table - no return springs

Y1	A+
Y2	B+
Y3	C+
Y4	D+

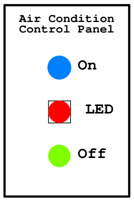
d. Output Table - with return springs



(no return springs)

e. PLC and I/O System

Fig. 3-9 PLC I-O Variables Assignment



Typical air-condition system is operated by two push-botton switched :

START button that operates the system

STOP button that deactivate the system

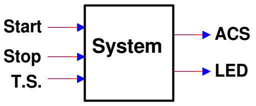
System is turned-on by pressing STARTbutton, and remains activated after releasing the button

System is tuened-off by pressing STOP button, and remains de-activated after releasing the button

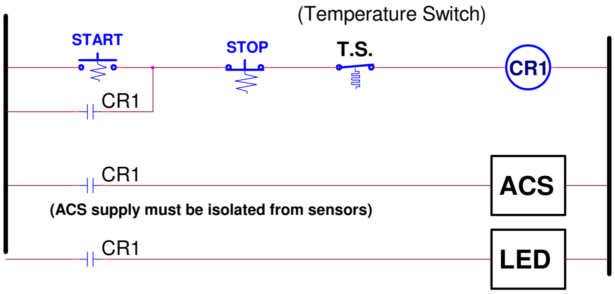
A red LED is illuminated as long as the system in on

In addition, control circuit is also equipped with a temperature sensor, that protects the system from over-heat (by turns it off)

a. Air-Condition System (ACS) representation

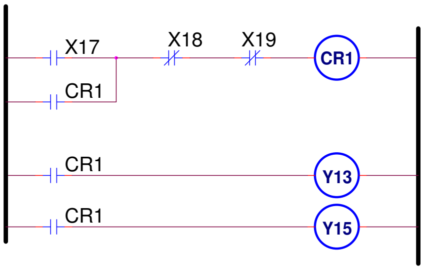


b. Block Diagram



(ACS supply must be isolated from sensors)

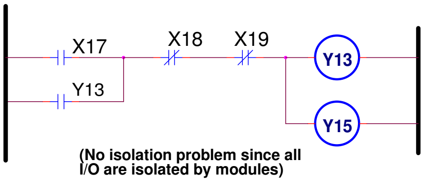
c. Relay Ladder Diagram for ACS



e. PLC "Internal" Ladder Diagram

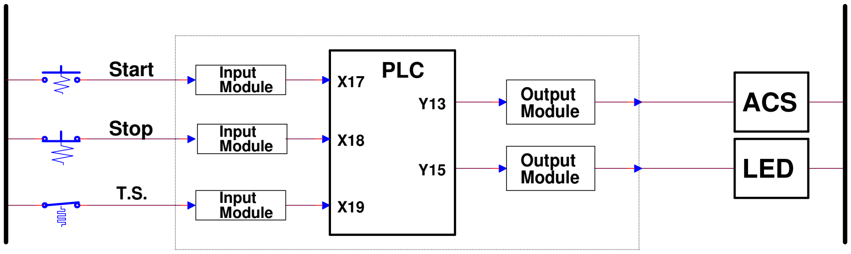
	INPUT MODULE	OUTPUT MODULE
Start	X17	
Stop	X18	
T.S.	X19	
ACS		Y13
LED		Y15

d. I/O Assignment Table



(No isolation problem since all I/O are isolated by modules)

f. Simplified Ladder Diagram

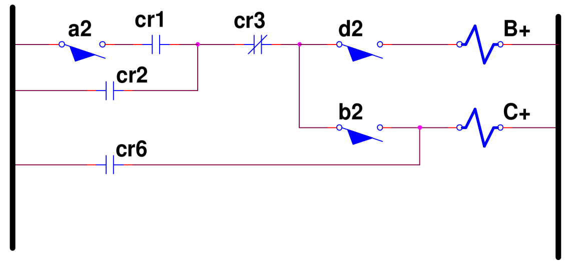


d. PLC Ladder Diagram

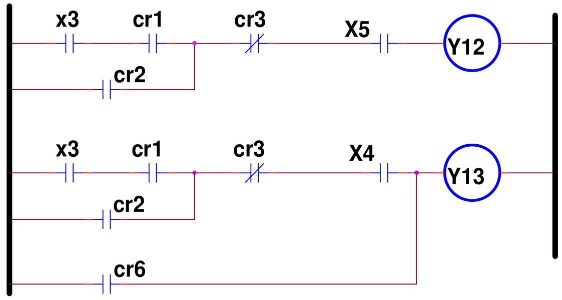
```
STORE          X17
OR             Y13
AND     NOT    X18
AND     NOT    X19
OUT            Y13
OUT            Y15
```

g. PLC Program

Fig. 3-10 PLC Control for Air-Condition System (ACS)



a. Partial Ladder Diagram of a Control System



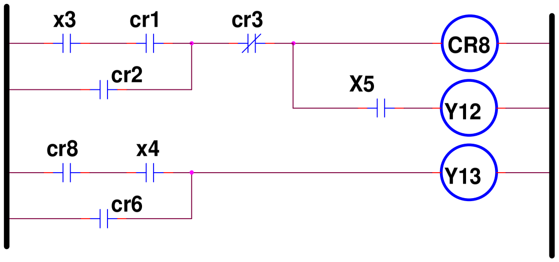
c. PLC Ladder Diagram

```
0    STORE      x3
1    AND        cr1
2    OR         cr2
3     AND NOT   cr3
4    AND        x5
5    OUT        Y12
6    STORE      x3
7    AND        cr1
8    OR         cr2
9     AND NOT   cr3
10   AND        x4
11   OR         cr6

12   OUT        Y13
```

d. PLC Program

Input Module	Output Module	Connected to
X3		a2
X4		b2
X5		d2
	Y12	B+
	Y13	C+

b. I/O Assignment



e. Simplified Ladder Diagram

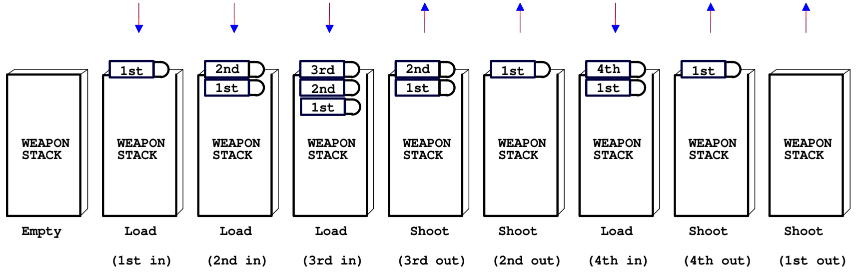
```
0    STORE      x3
1    AND        cr1
2    OR         cr2
3     AND NOT   cr3
4    OUT        CR8
5    AND        X5
6    OUT        Y12
7    STORE      cr8
8    AND        X4
9    OR         cr6
10   OUT        Y13
```

f. PLC Program
(simplified)

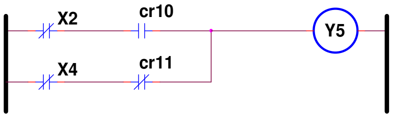
Fig. 3-11 : Simplification of PLC Ladder Diagram

Command STORE first stores current value in a stack, while "pushing down" previous stack elements, and then clears latest calculated value. Action is similar to loading weapon stack.

Commands that refer to stack - such as "AND STORE" , "OR STORE" etc - perform operation on current "upper" stack element (the one that been last entered), and deletes that element from stack, while "Pushing up" previous eelements. Action is similar to unloading a weapon stack, or shooting.



3-12 PLC LIFO Stack concepts

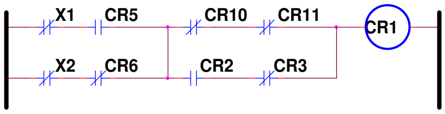


a. Ladder Diagram

```
0     STORE NOT    X2
1     AND          CR10
2     STORE NOT    X4
3     AND          CR11
4     OR           STORE
5     OUT          Y5
```

b. PLC Programming of (a)

Fig. 3-13 : Use of LIFO Memory Stack

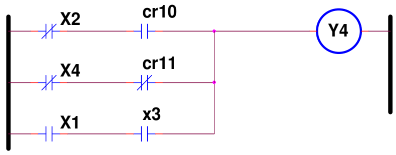


a. Ladder Diagram

```
STORE NOT    X1
AND          CR5
STORE NOT    X2
AND NOT      CR6
OR           STORE
STORE NOT    CR10
AND NOT      CR11
STORE        CR2
AND NOT      CR3
OR           STORE
AND          STORE
OUT          CR1
```

b. PLC Programming of (a)

Fig. 3-14 : Multiple Use of LIFO Stack



a. Ladder Diagram

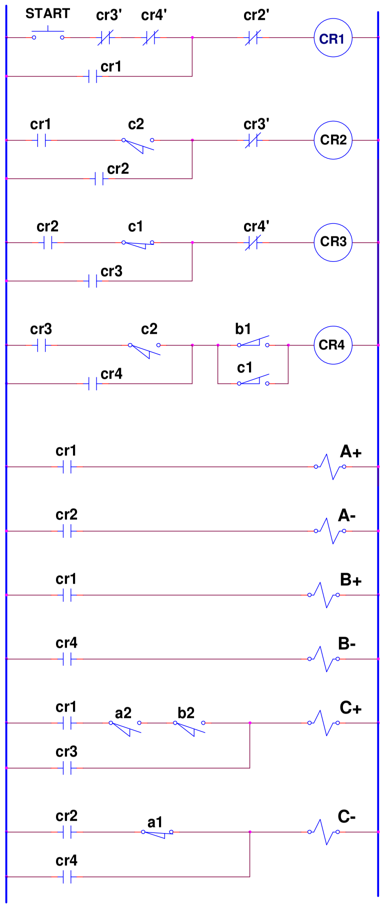
```
0     STORE NOT    X2
1     AND          CR10
2     STORE NOT    X4
3     AND          CR11
4     OR           STORE
5     STORE NOT    X1
6     AND          X3
7     OR           STORE
8     OUT          Y4
```

b. PLC Programming Option 1

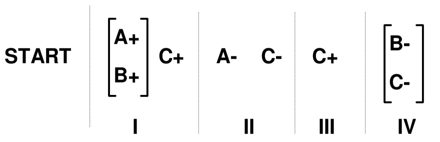
```
0     STORE NOT    X2
1     AND          CR10
2     STORE NOT    X4
3     AND          CR11
4     STORE NOT    X1
5     AND          X3
6     OR           STORE
7     OR           STORE
8     OUT          Y4
```

c. PLC Programming Option 2

3-15 : LIFO Use Oprions



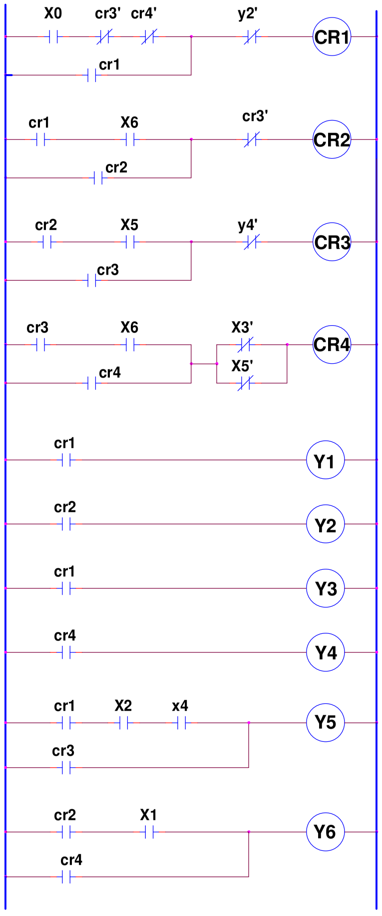
b. Relays Cascade Circuit



a. Process sequence Groups partitining

START	x0
a1	x1
a2	x2
b1	x3
b2	x4
c1	x5
c2	x6
A+	Y1
A-	Y2
B+	Y3
B-	Y4
C+	Y5
C-	Y6

c. PLC Variables Assignment

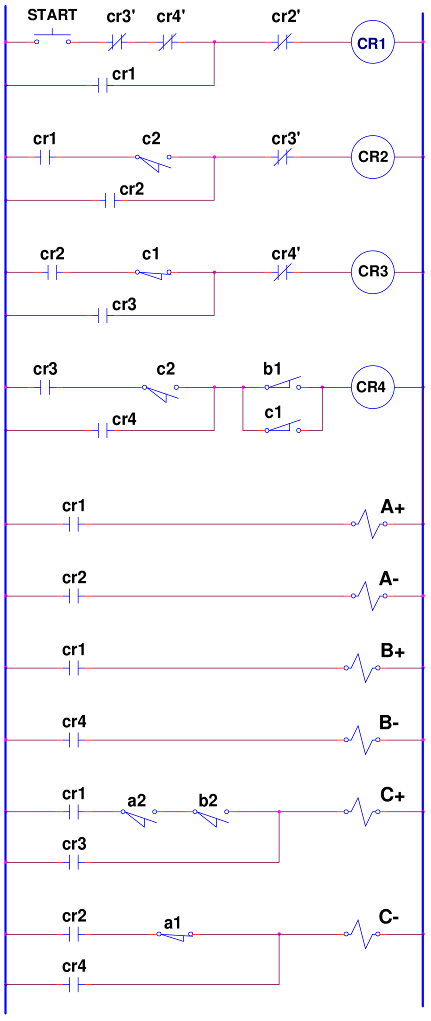


d. PLC Cascade Circuit

STORE	x0
AND NOT	cr3
AND NOT	cr4
OR	cr1
AND NOT	cr2
OUT	CR1
AND	x6
OR	cr2
AND NOT	cr3
OUT	CR2
AND	x5
OR	cr3
AND NOT	cr4
OUT	CR3
AND	x6
OR	cr4
STORE NOT	x3
OR NOT	x5
AND	STORE
OUT	CR4
STORE	cr1
OUT	Y1
STORE	cr2
OUT	Y2
STORE	cr1
OUT	Y3
STORE	cr4
OUT	Y4
STORE	cr1
AND	x2
AND	x4
OR	cr3
OUT	Y5
STORE	cr2
AND	x1
OR	cr4
OUT	Y6

e. PLC Program

Fig. 3-16 : PLC Cascade System Design



b. Relays Cascade Circuit

START [A+ B+] C+ | A- C- | C+ | [B- C-]

I | II | III | IV

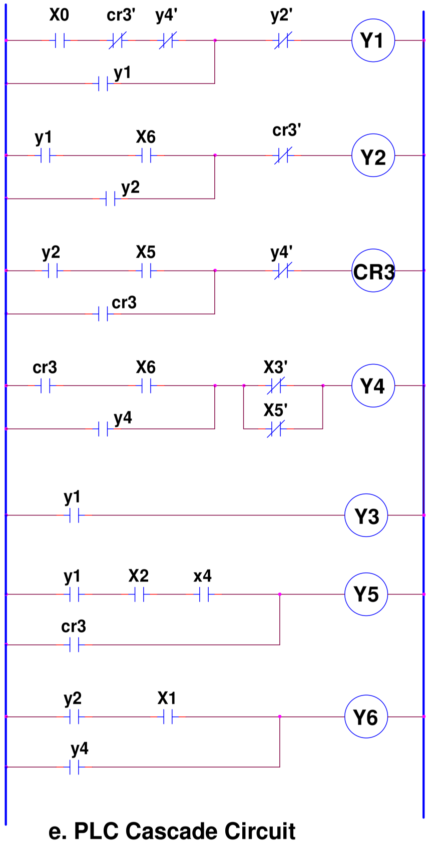
a. Process sequence Groups partitining

START	x0
a1	x1
a2	x2
b1	x3
b2	x4
c1	x5
c2	x6
A+	Y1
A-	Y2
B+	Y3
B-	Y4
C+	Y5
C-	Y6

c. PLC Variables Assignment

Y1 = CR1
Y2 = CR2
Y4 = CR4

d. Optimize CR Relays



e. PLC Cascade Circuit

```
STORE      X0
AND NOT    CR3
AND NOT    Y4
OR         Y1
AND NOT    Y2
OUT        Y1
AND        X6
OR         Y2
AND NOT    CR3
OUT        Y2
AND        X5
OR         CR3
AND NOT    Y4
OUT        CR3
AND        X6
OR         Y4
STORE NOT  X3
OR NOT     X5
AND        STORE
OUT        Y4

STORE      Y1
OUT        Y3
AND        X2
AND        X4
OR         CR3
OUT        Y5
STORE      Y2
AND        X1
OR         Y4
OUT        Y6
```

f. PLC Program

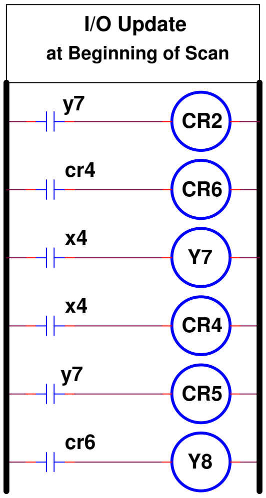
Fig. 3-17 : PLC Simplified Cascade of Fig. 3-16

1. Read & store status of all external inputs (Xi)
2. Run program commands in its written order
3. Wait until end of scan time (*)
4. Return to (1) and start new cycle

a. PLC Scan Cycle Order

1. Input variables (Xi) status cannot be changed by program commands, and remains unchanged until next cycle
2. Output variables (Yi) and internal variables (CRi) status are controlled by PLC program. They are updated once during each cycle
3. Program commands refer to latest update of the variables.

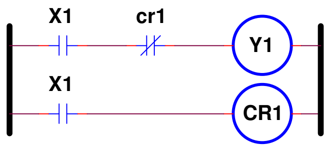
b. Variables Status During Scan Cycle



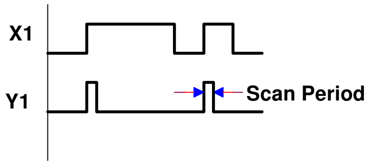
Variables States	
This Scan	Next Scan
Y7=0 , CR2=0	Y7=1 , CR2=1
CR4=0 , CR6=0	CR4=1 , CR6=1
X4=1 , Y7=1	
X4=1 , CR4=1	
Y7=1 , CR5=1	
CR6=0 , Y8=0	CR6=1 , Y8=1

c. Effect of Scanning Action on Relay States

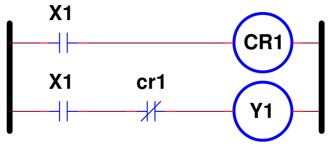
Fig. 3-18 : Effect of Scanning Action in PLC



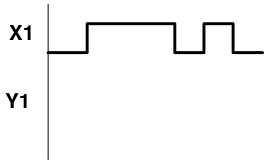
```
STORE      X1
 AND NOT   CR1
 OUT        Y1
STORE      X1
 OUT       CR1
```



a. Proper Rung Order

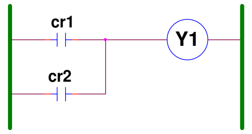
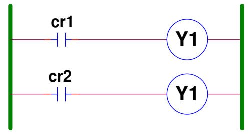


```
STORE      X1
 OUT       CR1
 AND NOT  CR1
OUT         Y1
```

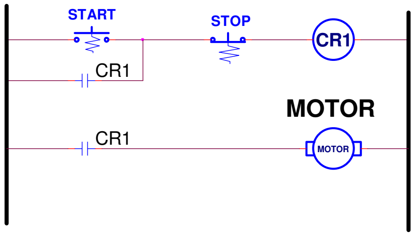
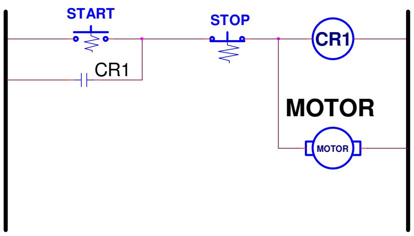


b. Improper Rung Order

Fig. 3-19: Importance of Proper Rung Order (Pulse Shaper)

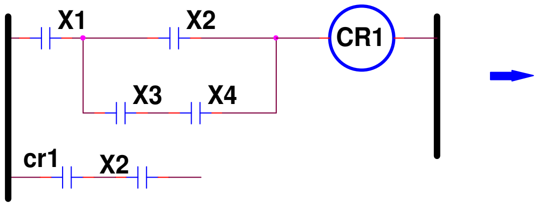
RIGHT	WRONG	
		**SCHEMATIC DRAWING**
STORE cr1 OR cr2 OUT Y1	STORE cr1 OUT Y1 STORE cr2 OUT Y1	**PLC PROGRAM**
CR1=1 CR2=0	CR1=1 CR2=0	**SPECIFIC CASE**
STORE cr1 1 OR cr2 1+0=1 OUT Y1 Y=1	STORE cr1 1 OUT Y1 Y=1 STORE cr2 0 OUT Y1 Y=0	**PROGRAM FLOW**
Y=1	Y=0	**END-LOOP RESULT**

a. NEVER SPLIT OUTPUT FUNCTION INTO 2 (OR MORE) OUTPUT SUB-FUNCTIONS

RIGHT	WRONG
	
Motor current flows through separate contact, thus isolated from input sensors	Motor current flows through sensors)

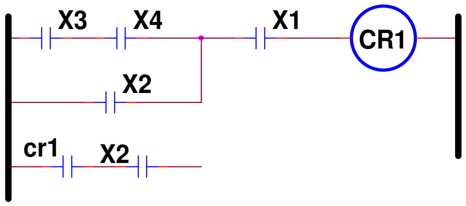
b. LOAD MUST BE ISOLATED FROM SENSORS

Fig. 3-20 : Right/Wrong Cases



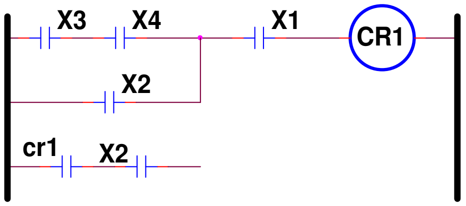
STORE	X1
STORE	X2
STORE	X3
AND	X4
OR	STORE
AND	STORE
OUT	CR1
STORE	cr1
AND	X2

a. Original Circuit



STORE	X3
AND	X4
OR	X2
AND	X1
OUT	CR1
STORE	cr1
AND	X2

b. Programmable Simplified Circuit



c. Ladder Sub-Circuit

STORE	X3
AND	X4
OR	X2
AND	X1
OUT	CR1
STORE	cr1
AND	X2

d. PLC Program

STORE	X3
AND	X4
OR	X2
AND	X1
OUT	CR1
AND	X2

e. Simplified Program

Fig. 3-21 : PLC Program Simplification

TIMERS AND COUNTERS

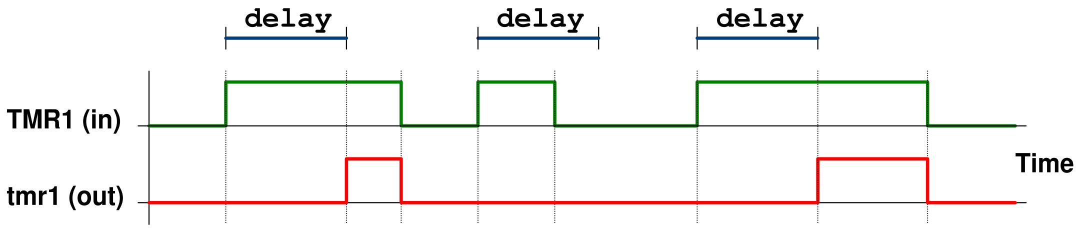
The following implement PLC timers and counters of two companies :

General Electrinc (GE)

Texas Instruments (TI)

Note : Examples refer to specific PLCs. Actually there exist a lot of different timers and counters.

Fig. 3-22 : PLC Timer And Counter Models



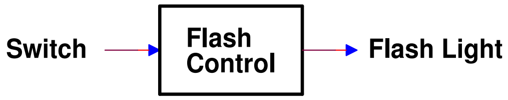
a. "On-Delay" Definition

X1=1 Timer Enabled and counting time
X1=0 Timer is Reset

b. Specific Case

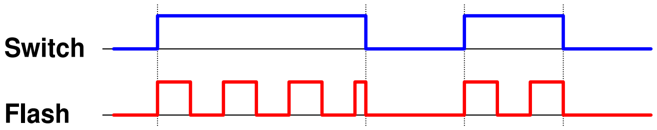
c. General Case

Fig. 3-23 : PLC "On-Delay" Timer Model (GE)

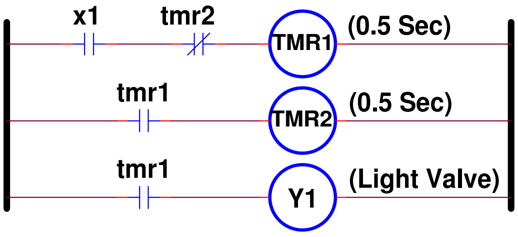


a. Flash Block Diagram

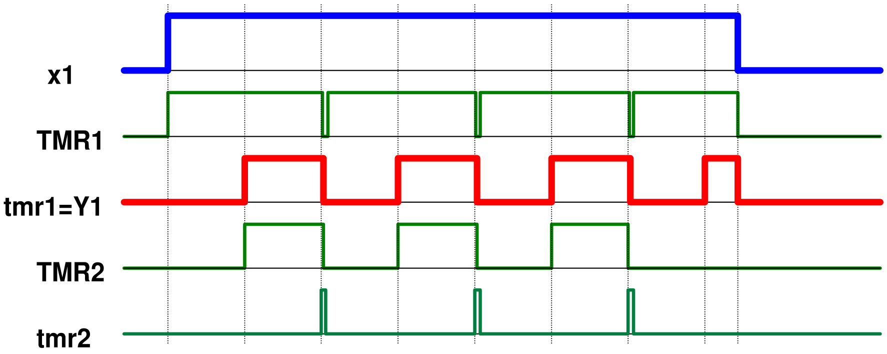
Flash light is actuated while switch is on



b. Flash Timing Diagram

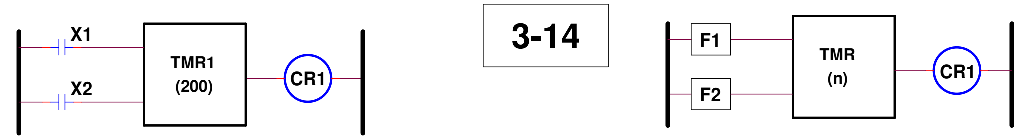


c. PLC Ladder Diagram



d. Detailed Timing Diagram

Fig. 3-24 : Design Flash-Light system using "On-Delay" time



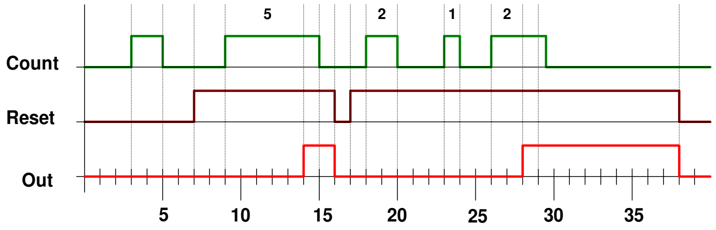
X1=1 Timer Run X1=0 Timer Stops
X2=1 Timer Enabled X2=0 Timer is Reset

F1=1 Timer Run F1=0 Timer Stops
F2=1 Timer Enabled F2=0 Timer is Reset

```
        STORE    X1
        STORE    X2
        TMR1
empty   200
        OUT      CR1
```

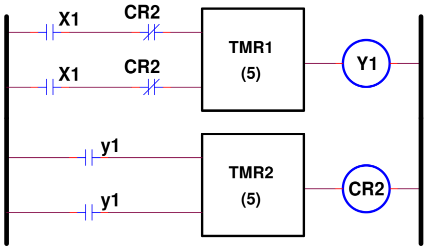
a. Specific Case

b. General Case



c. Timing Diagram (Delay=0.5 Sec)

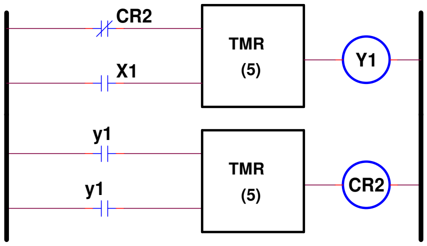
Fig. 3-25 : 2-Input PLC Timer (TI - Texas Instruments)



```
        STORE      X1
        AND NOT    CR2
        STORE      X1
        AND NOT    CR2
        TMR1
empty   5
        -
        OUT        y1
        STORE      y1
        STORE      y1
        TMR2
empty   5
        -
        OUT        CR2
```

a. Ladder Diagram

b. Programming of (a)



```
        STORE NOT   CR2
        STORE       X1
        TMR1
empty   5
        -
        OUT         Y1
        STORE       y1
        STORE       y1
        TMR2
empty   5
        -
        OUT         CR2
```

c. Simplified Ladder Diagram

d. Programming of (c)

Fig. 3-26 : Design Flash-Light system using TI timers

A security lock is to be opened only on confirmation of two managers, simultaneously. Each manager confirms by turning on a personal switch (A and B).

In order to make sure that a single manager will not be able to turn on both switches, the switches been located far from each other, and lock is to be opened only if the two switches are turned on within 0.5 Sec.

If one switch is turned on at least 0.5 Sec before second switch is turned on, the security lock is disabled, and may later be open after releasing the pressed switch and repeat the process.

After being opened, the lock is closed as soon as any of the switches is turned off.

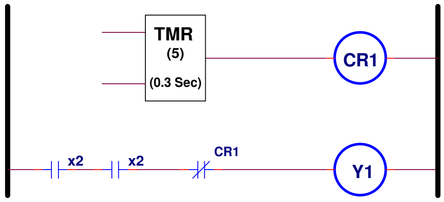
a. Security Lock Specifications

Switch A Switch B	X1 X2
Lock	Y1

b. PLC I/O table

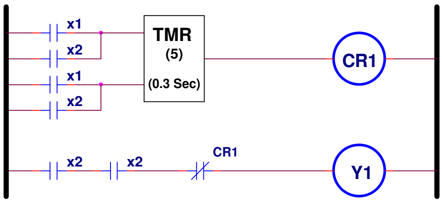
Y=1 when both switches are operated
Missing timing dependence

c. Solution Step 1



Y=1 when both switches are operated, and timer output tmr1 is off
Missing timer actuation

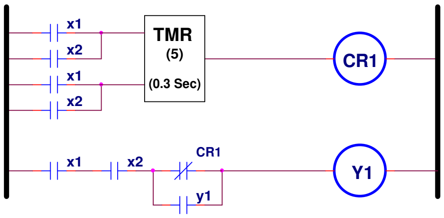
d. Solution Step 2



Timer is actuated by either of the two switches. If timeout occurs before seconad swithced been turned, lock is diabled

Timer output always is set to "1" after 0.5 Sec. This disables lock even if it had been opened correctly.

e. Solution Step 3



If Y1 been set to "1", it keeps be connected by by-passing cr1 contact (acts as flip-flop).

f. Solution Step 3 - Complete Solution

Fig. 3-27 : Two-Hands Circuit Using PLC

X1= O-->1 One count
X2=1 Counter Enabled X2=0 Counter is Reset

a. Specific Case

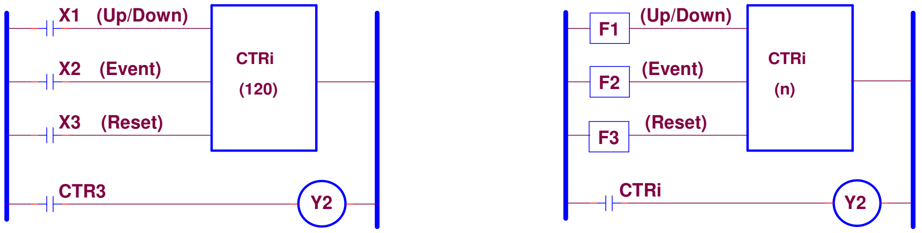
F1= O-->1 One count
F2=1 Counter Enabled F2=0 Counter is Reset

b. General Case

```
        STORE    X1
        STORE    X2
        CTR1
empty—→ 25
        OUT      CR1
```

c. Programming of (a)

Fig. 3-28 : PLC Two-Input UP Counter (TI)



X1 determines UP or DOWN mode
X2= O-->1 One count
X3=1 Counter Enabled
X3=0 Counter is Reset

a. Specific Case

F1 determines UP or DOWN mode
F2= O-->1 One count
F3=1 Counter Enabled
F3=0 Counter is Reset

b. General Case

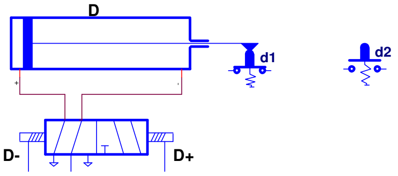
```
LOAD NOT   X1
LOAD NOT   X2
LOAD NOT   X3
OUT        CTR3
120
 LOAD      CTR3
OUT        Y1
```

c. Programming of (a)

Fig. 3-29 : 3-Input UP/DOWN Counter (GE)

START,(D+,D-) repeat 6 times,D+,10 Sec DELAY,D-

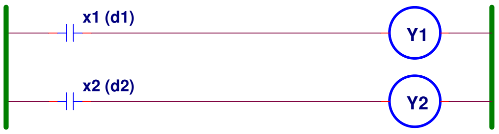
a. Required Sequence



b. Cylinder Type

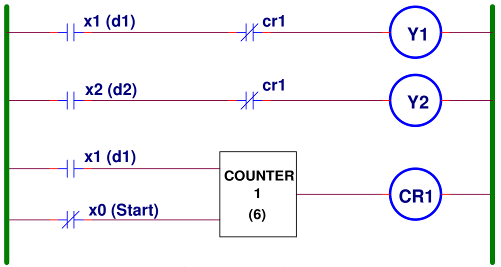
Input Modules	Output Modules	Connected to
X0		START
X1		d1
X2		d2
	Y1	D+
	Y2	D-

c. PLC I/O Table



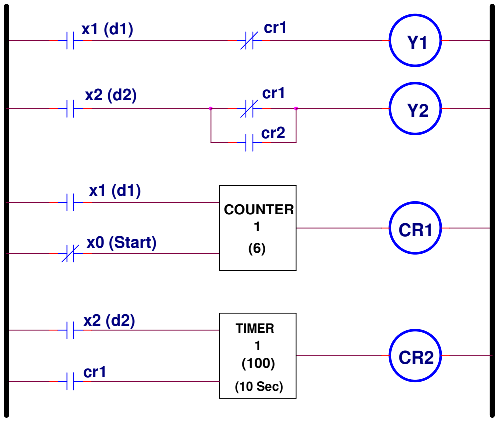
d. Solution Step 1

Infinite cycles



e. Solution Step 2

Circuit performs 6+1/2 Cycles, and stops



f. Solution Step 3 - Complete Circuit

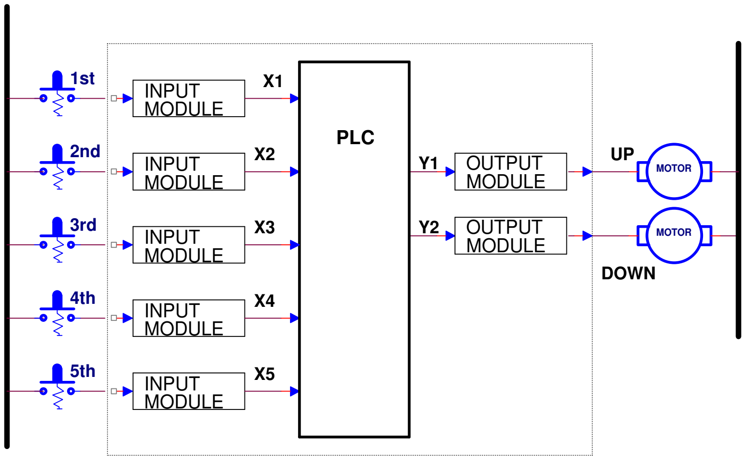
STORE	x1
AND NOT	cr1
OUT	Y1
STORE NOT	cr1
OR	cr2
AND	x2
OUT	Y2
STORE	x1
STORE NOT	x0
CTR1	
6	
(NOT USED)	
OUT	CR1
STORE NOT	x2
STORE	cr6
TMR1	
100	
(NOT USED)	
OUT	CR2

g. PLC Programm

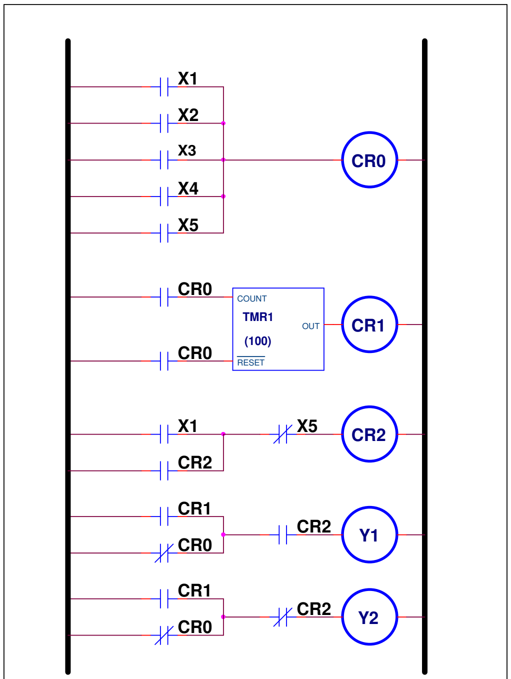
Fig. 3-30 : Step-By-Step Design of Sequence
START,(D+,D-) repeat 6 times,D+,10 Sec DELAY,D-

An elevator exists in a 5-floor building.
Elevator runs automatically (without human control), from 1st floor to 5th floor, and vica versa, while stopping for 10 seconds in each floor.
Each floor is equipped WITH a unique contact, that is closed when the elevator reaches that floor.
These contacts are implemented as input sensroes for PLC control system.

a. System Description



b. PLC System Configuration



Notes : CR2 control elevator direction
Y1 - Up direction
Y2 - Down direction

c. PLC Ladder Diagram

STORE	X1
OR	X2
OR	X3
OR	X4
OR	X5
OUT	CR0
AND	CR0
TMR1	
100	
(NOT USED)	
OUT	CR1
STORE	X1
OR	CR2
AND NOT	X5
OUT	CR2
STORE	CR1
OR NOT	CR0
AND	CR2
OUT	Y1
STORE	CR1
OR NOT	CR0
AND NOT	CR2
OUT	Y2

d. PLC Program

Fig. 3-31 : "Saturday" Elevator

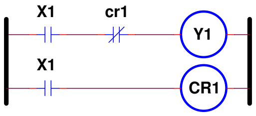
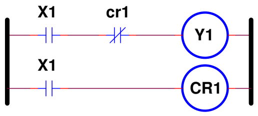
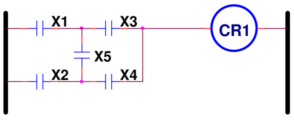
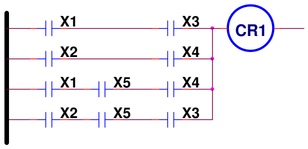
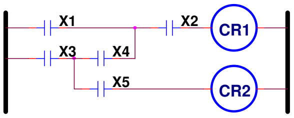
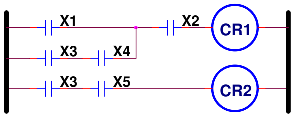
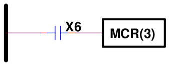
Property	Relays System	PLC System
a. Signal Flow	Parallel	Serial
b. Number of Contacts	Limitted	Unlimitted
c. Adding Relay	Expenssive	NO cost
d. Timers / Counters	Expenssive	NO cost
e. Reliability	Limmted	Almost unlimmited
f. Support	Required	Almost not required
g. Races	 Possible	 Impossible
h. Non Serial Parallel (CR1 = X1.X3+X2.X4+ +X1.X5.X4+X2.X5.X3)	 Possible	 Impossible
i. Sneak Path	 CR1 = X1.X2 + X3.X4.X2 CR2 = X3.X5 + X1.X4.X5 Possible	 CR1 = X1.X2 + X3.X4.X2 CR2 = X3.X5 Impossible

Fig. 3-32 :Differences Between Relay and PLC Ladder Diagram

Byte/Word format (binary groups)
Digital to/from Binary
Mathematical Operations
Matrix Operations
Digital/Analog conversion
More

Fig. 3-33 : Support Other Data Types



a. Ladder Diagram

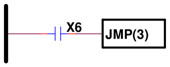
```
      ⋮
STORE   X6
MCR     3
      ⋮
```

b. Programming Lines

If X6=0, then the next (3) outputs are kept at logic "0" (shut off)

c. Command Definition

Fig. 3-34 : Master Control Relay (MCR) Command



a. Ladder Diagram

```
      ⋮
STORE   X6
JMP     3
      ⋮
```

b. Programming Lines

If X6=0, then the next (3) outputs are frozen, and the PLC continues its scan at the rung following the third output

c. Command Definition

Fig. 3-35: Jump (JMP) Command

ENHANCED LADDER DIAGRAM INSTRUCTIONS

Enhanced ladder diagram instructions are handeled very differently in different PLC models. Therfore, only a few typical examples are shown here.

LATCH (UNLATCH)
GOTO
ADD
SUBTRACT
MULTIPLICATION
DIVISION
SQUARE ROOT
COMPARE

Fig. 3-36 : Arithmetic Instruction

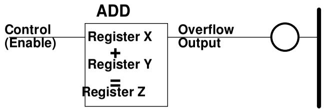


Fig. 3-37 : Typical "ADD" Instruction Block

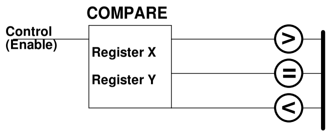


Fig. 3-38 : Typical "COMPARE" Instruction Block

AND
OR
XOR
NAND
NOR
NOT

Fig. 3-39 : Logic Operations

(or correcponding bits of two specified registers)

These change the contenta of a specified register.

Typical Conversions :

BCD to Binary
Binary to BCD
Absolute Value
Complement (multiplied by -1)
Invert (inverts all bits)

3-40 Data Conversion Instructions

Logical Shifts : Moves all bits of a register to right or left

Data Transfer Instructions :
There are many such instructions, used to move data within the PLC

Fig. 3-41 : Data Commands

MICRO-AUTOMATION versus MACRO-AUTOMATION

Micro-Automation :	Automation of a single machine, or a small group of machines, also called an "Automation Island".
Macro-Automation :	Automation of a large plant, consisting of many individual automation islands. Uses a large computer or a large PLC, that coordinates the operation of the individual automation islands. Each automation island is uusually controlled by its own control system, such as a small or medium-scale PLC ("Distributed Control").

Macro-automation makes use of a LAN (Local-Area-Network), which links the various stations or automation islands with the central supervising computer or PLC. The use of a LAN provides :

1. Distributed Control
2. Centralized Data Acquisition

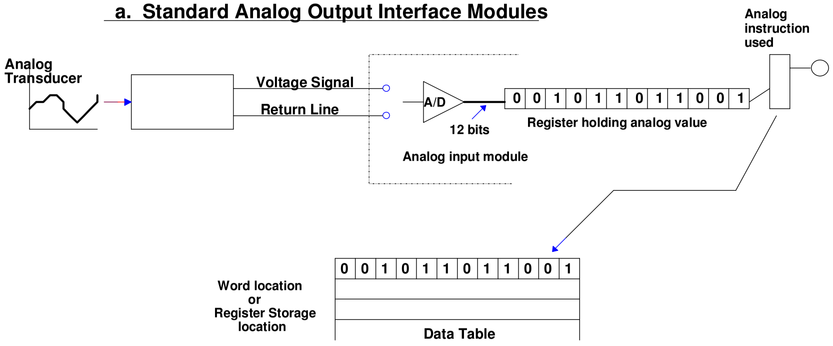
The LAN can be organized according to any one of a several possible LAN Topologies (arrangements), such as "Common Bus", "Star" or "Ring".

THIS COURSE REFERS TOMICRO-AUTOMATIC ONLY

Fig. 3-42 : PLC Micro/Macro Systems Definition

4 - 20 mA
10 - 50 mA
0 - 5 volt DC
0 - 10 volt DC
+/- 5 volt DC
+/- 10 volt DC
PID Controller

a. Standard Analog Output Interface Modules



Since A/D converters are expensive, they are often shared by several 4 to 8 channels. One channel is read and stored on every scan.

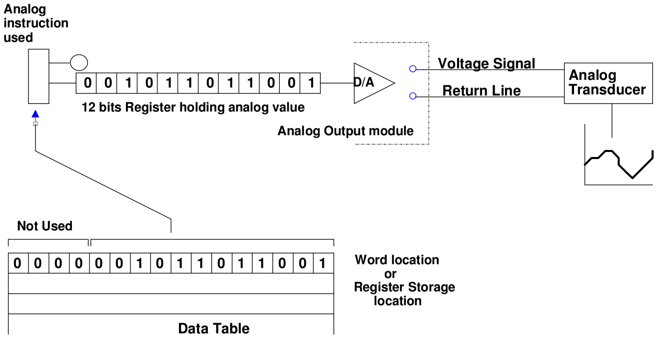
b. Storing an Analog Input

ANALOG IN

Rack o Slot 03	(Rack with 8 slots) (Slots 0 to 7 per rack)
Number of channels 8 Destination register 200	(Octal code 200 = 128)

(Since there are 8 channels, the 8 slots would be assigned the Destination Register 200 ^{8}to 207 §

c.. Typical "analog In" block Instruction



d. Tranferring a Word to Analog Output Module

ANALOG OUTPUT

Rack 4 Slot 05	(Rack with 4 slots) (Slots 0 to 7 per rack)
Number of channels 4 Source register 300	(Octal code 300 = 192)

e. Typical "Analog Out" Instruction Block

PID CONTROLLER

Control (Enables Outputs)	Connected to Register	
	1. Controlled Variable	(Given by analog input module)
	2. Controller Output	(Sent to analog output module)
Track (Tracks incoming variables, even if "Control" is off)	3. K (P-control)	○ (Active loop control)
	4. R (I-control)	○ (High alarm limit)
	5. T (D-control)	○ (Low alarm limit)
	6. Set Point	

In some PLC's, the PID Controller is implemented by a special output module, i.e. by hardware. In others, it is programmed using a special Insrtuction Block, such as shown here.

f. Typical "PID Controller" Instruction Block

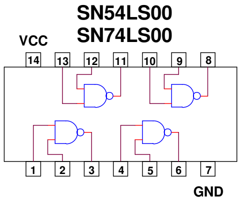
Fig. 43 : Support Analog Data

CHAPTER 4

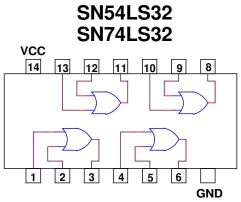
INDUSTRIAL SWITCHING ELEMENTS

Electronic Gates
Gates Implementation
Input/Output Module
Electric Relays
Pneumatic Valve “Gates”
Moving Parts Logic (MPL)
Design Considerations
Switching Elements Properties

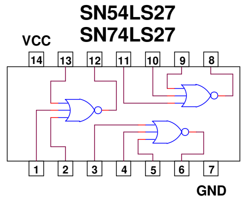
4-1



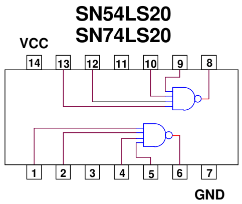
a. Quad 2-Input NAND Gate



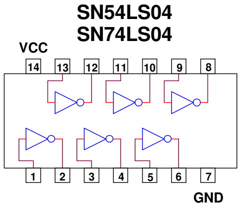
b. Quad 2-Input OR Gate



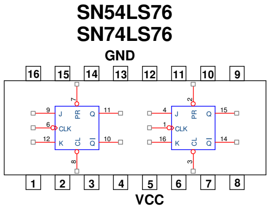
c. Tripple 3-Input NAND Gate



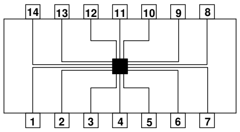
d. Dual 4-Input NAND Gate



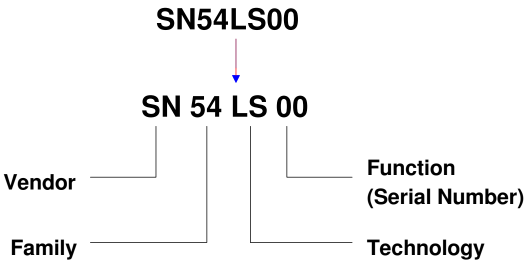
e. Hex INVERTER



f. Dual J-K FLIP-FLOP



g. Chip Configuration



h. Chip Name "Code"

74 : Commercial Family
54 : Military Family

Difference : Temperature Range

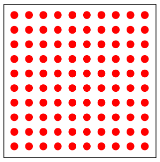
Many More Families

i. Families

High / Low Speed
High / Low Power Dissipation
High / Low Power Drive (Fan-Out)
High / Low Power Supply Voltage

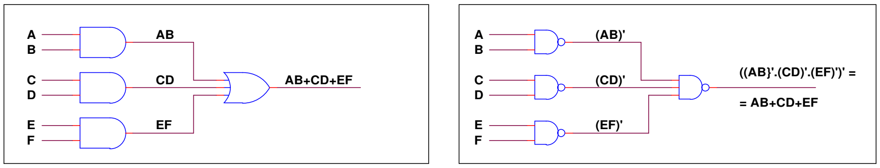
And Many More

j. Technologies Characteristics



k. Grid-Array Package
(not SSI)

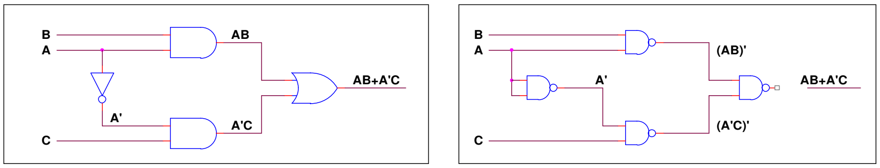
Fig. 4-1 : Typical Simple Gates Packages (SSI - Small Scale Integrated)



a. AND-OR-NOT system

b. NAND system

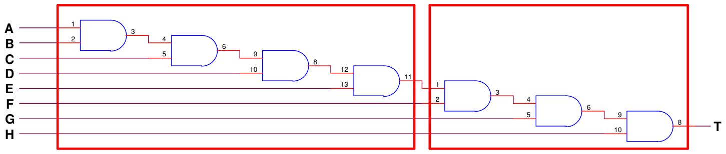
Fig. 4-2 : Equivalent Circuits for Function AB+CD+EF



a. AND-OR-NOT system
4 Gates , 3 Packages

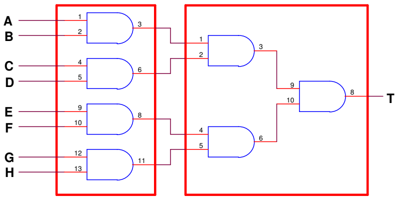
b. NAND system
4 Gates , 1 Package

Fig.4- 3 : Equivalent Circuits for Function AB+A'C (Save Packages)



T = ABCDEFGH

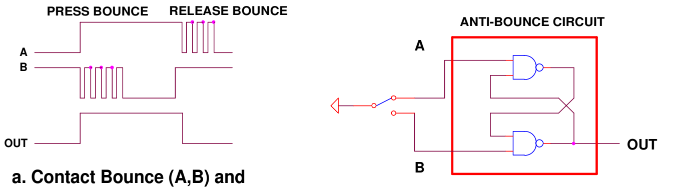
a. 7 Gates , 2 Packages , 7xT delay



b. 7 Gates , 2 Packages , 3xT delay

Fig. 4-4 : Equivalent Circuits for Function ABCDEFGH

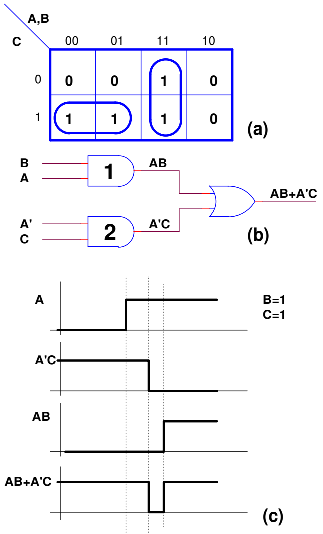
(Increase Speed)



a. Contact Bounce (A,B) and
Required Signal (OUT)

b. Eliminate Bounce by a Simple Flip-Flop

Fig. 4-6 : Switch Bounce Elimination



Transition From 011 to 111
(Gate 1 Slower Than Gate 2)

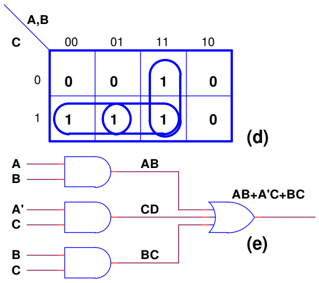


Fig. 4-5 : Hazard Elimination

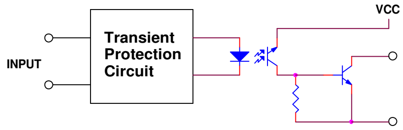
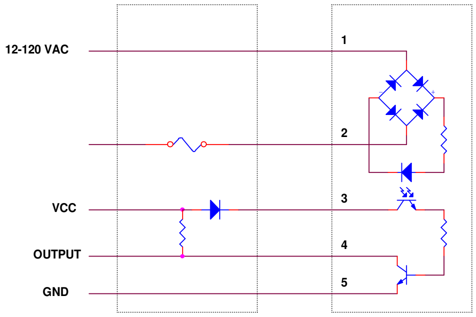
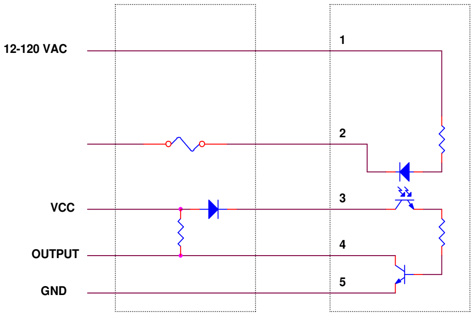


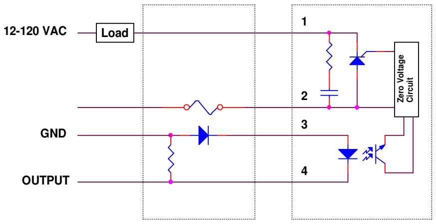
Fig. 4-7 : Typical Opto-Isolated Input Module



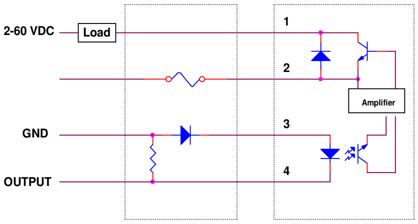
a. AC Input Module (IAC)



DC Input Module (IDC)



b. AC Output Module (IAC)



DC Output Module (IDC)

Fig. 4-8 : AC Input and Output Modules for Electronic Gates

Fig. 4-9 : DC Input and Output Modules for Electronic Gates

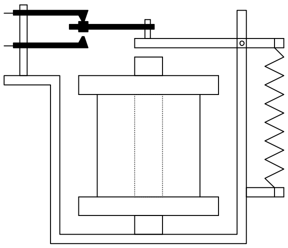


Fig. 4-10 : Relay Construction

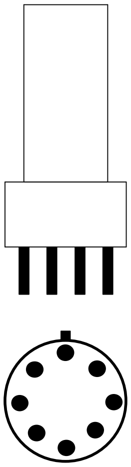


Fig. 4-11 : Typical Relay Package

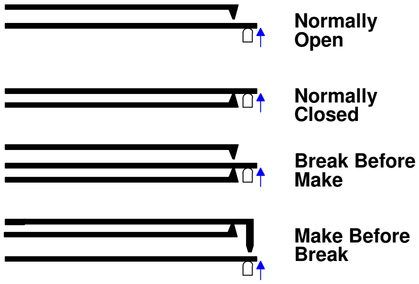
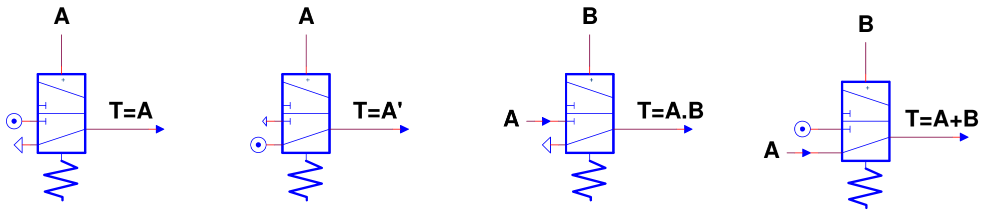
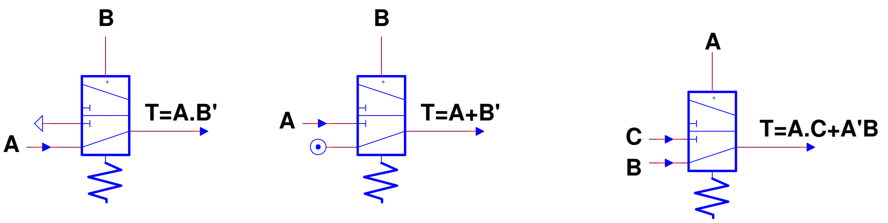


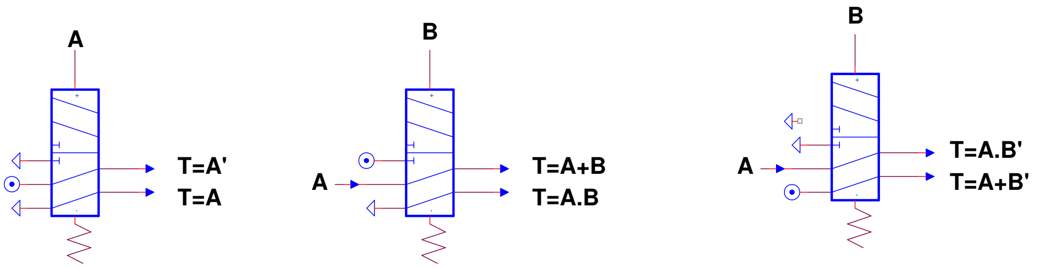
Fig. 4-12 : Typical Relay Contacts



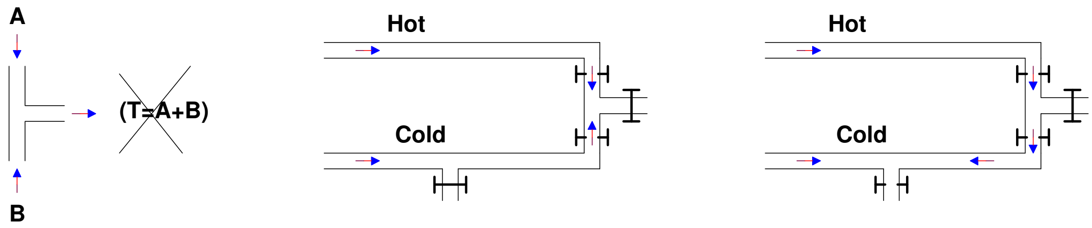
a. YES Gate b. NOT Gate c. AND Gate d. OR Gate



e. INHIBITION Gate f. IMPLICATION Gate g. SELECTOR Gate

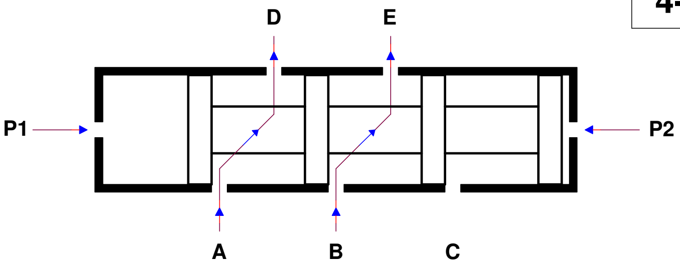


h. YES/NOT Gate i. AND/OR Gate j. INHIBITION/IMPLICATION Gate

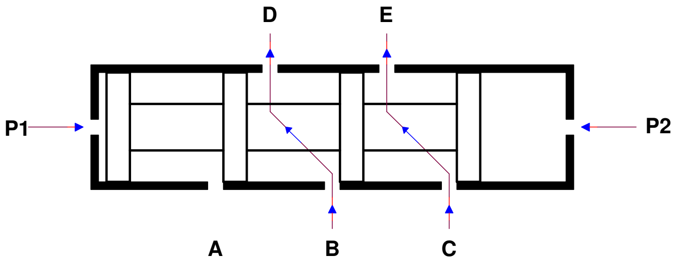


k. Non-Valve "OR" Element l. Almost equal HOT/COLD pressure - No Flow m. COLD Pressure Less Than HOT - HOT Flows Into COLD

Fig. 4-13: Pneumatic Valve Gates



a. Position 1



b. Position 2

Fig. 4-14 : Operation of 5/2 Spool Valve

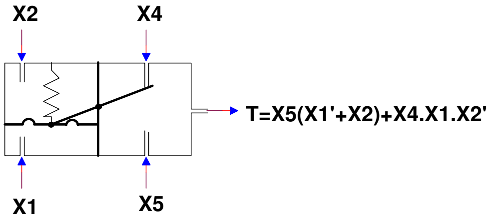
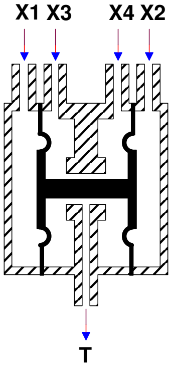


Fig. 4-16 : Pneumatic Universal Gate (MPL)
(Samsomatic, W. Germany)



T=X3(X1'+X2)+X4(X1+X2')

Fig. 4-17 : Pneumatic Universal Gate (MPL)
(Dreloea, E.Germany)

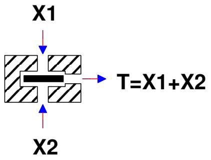


Fig. 4-15 : Shuttle Valve
(OR Gate)

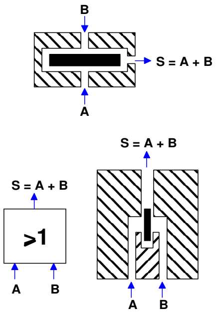


Fig. 4-18 : Pneumatic OR Gates

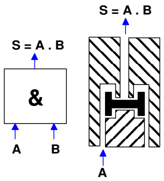
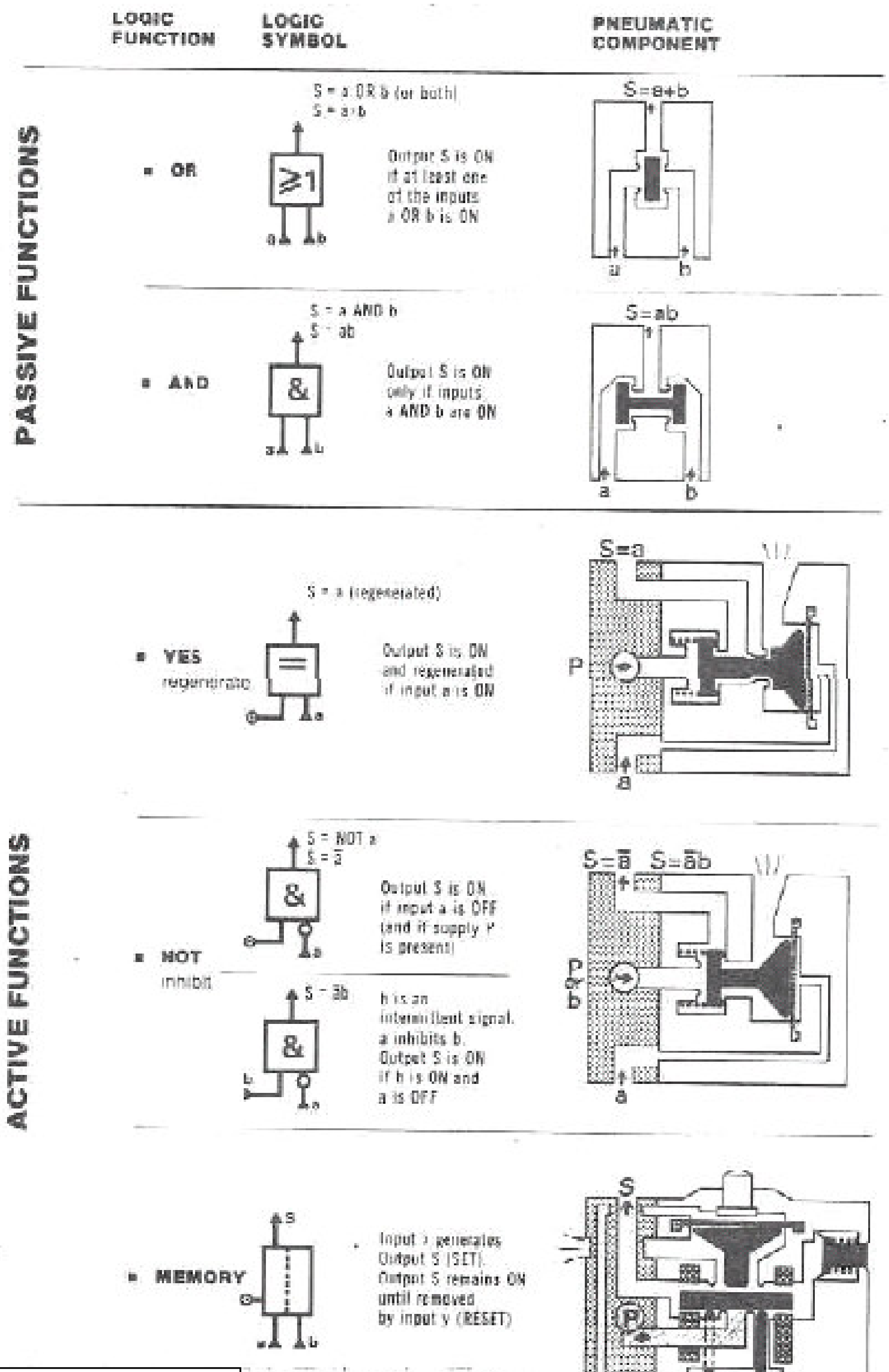


Fig. 4-19 : Pneumatic AND Gate



Output S is ON if at least one of the inputs a OR b is ON

Output S is ON only if inputs a AND b are ON

Output S is ON and regenerated if input a is ON

Output S is ON if input a is OFF (and if supply P is present)

b is an intermittent signal. a inhibits b. Output S is ON if b is ON and a is OFF

Input x generates Output S (SET). Output S remains ON until removed by input y (RESET)

Pneumatic Gates (TELEMECANIQUE, France)

Fig. 4-20 : Pneumatic Gates (TELEMECHANIQUE, France)

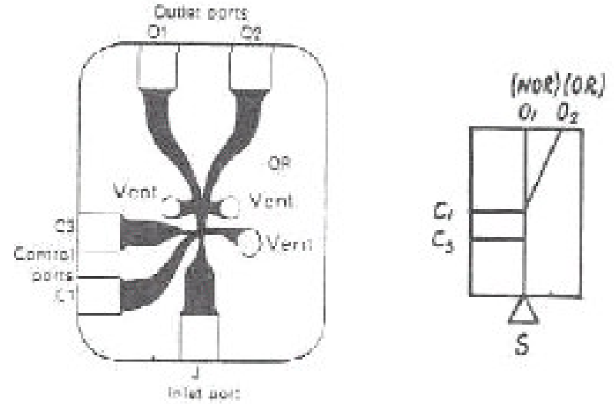


Fig. 4-21 : Fluid Flip-Flop (Wall Attachment Principle)

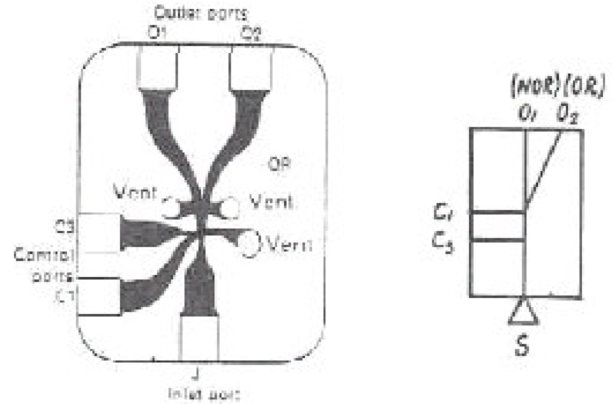
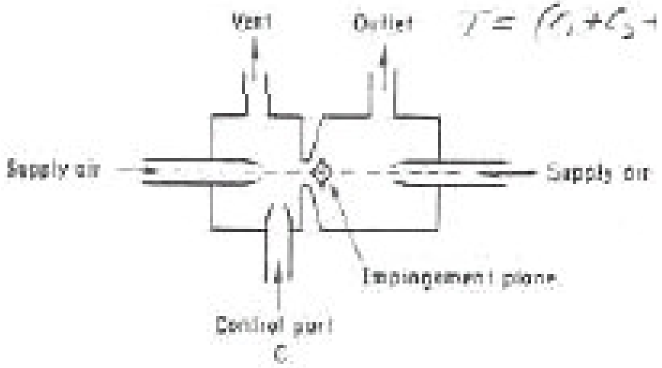


Fig. 4-22 : Fluid OR-NOR Gate **Q1 = C1 + C3 , Q1 = (C1 + C3)’**



$T = (C_1 + C_2 +$

Fig. 4-23 : Fluid NOR Gate (Impact Modulator, AIR Logic, USA)
T = (C1 + C2 + C3 + C4)’

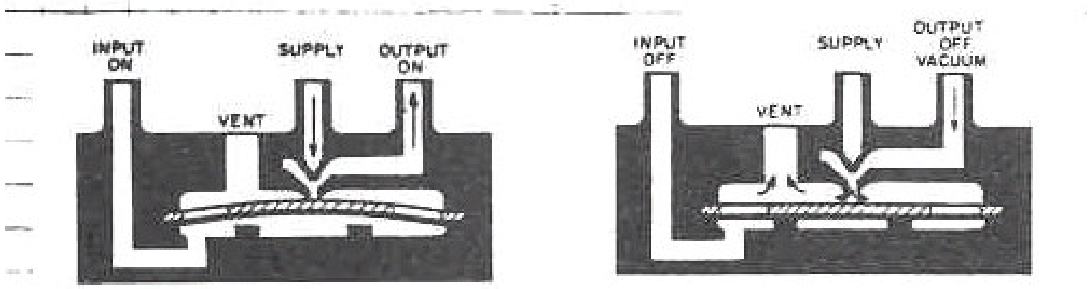


Fig. 4.33. *On* and *Off* states of a diaphragm amplifier. (Courtesy of Air Logic Div., Fred Knapp Eng. Co.)

Fig. 4-24 : Pneumatic Binary Amplifier (Single Stage, AIR LOGIC, USA)

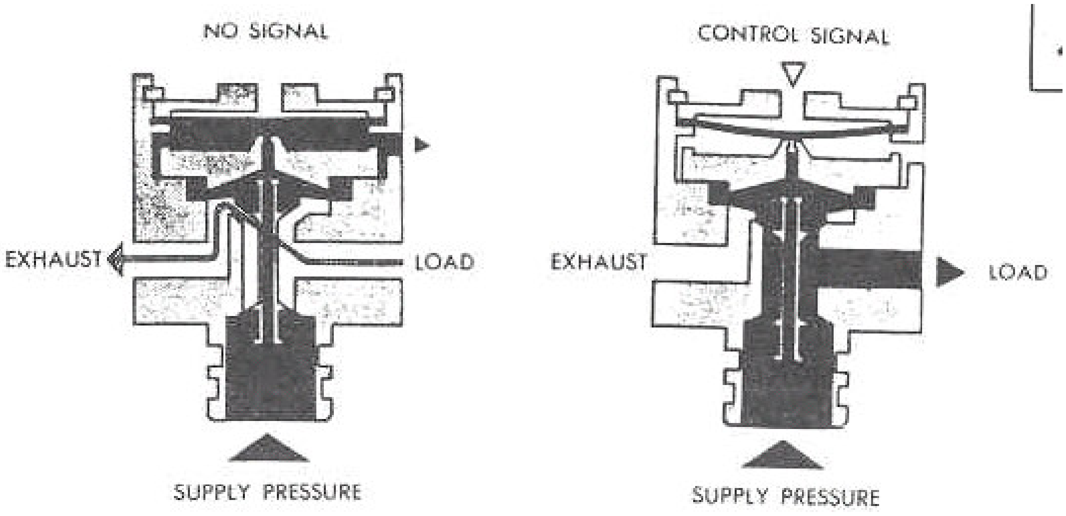


Fig. 4-25 : Pneumatic Binary Amplifier (Two Stage, CLIPPARD, USA)

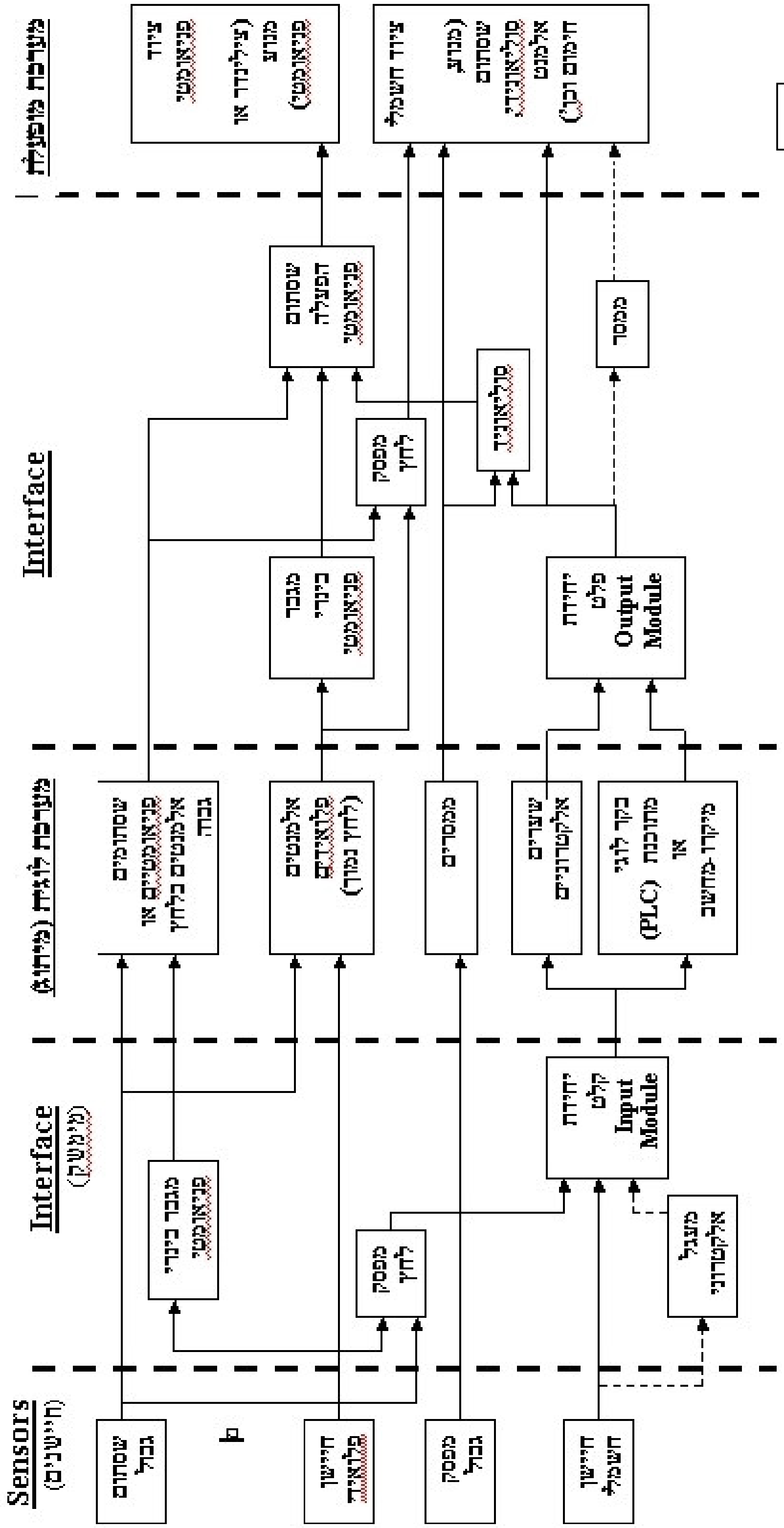


Fig. 4-26 : Comparison Between Switching Methods

	Electronics	Relays	Pneumatic Valves	Moving-Part Logic	Fluidies
זמן תגובה	5-10 nSec	10-25 mSec	10-20 mSec	5-10 mSec	1-2 mSec
גודל יחסי של המערכת	קטן	גדול	גדול	בינוני	בינוני
אמינות (משך חיים)	בלתי מוגבל בתנאים מתאימים	10E5-10E6 מחזורים	10E7-10E8 מחזורים	10E7-10E8 מחזורים	בלתי מוגבל עם אויר נקי
רגישות לטמפרטורה	גרוע	בינוני	טוב	טוב	טוב מאד
רגישות ללכלוך וקורוזיה	בינוני	בינוני	טוב	טוב	גרוע
רגישות לתנודות והלם	בינוני	גרוע	טוב מאד	בינוני	טוב מאד
רגישות לרעש חשמלי וקרינה	גרוע	טוב	טוב מאד	טוב מאד	טוב מאד
סכנה עם חומר מתלקח	בינוני	גרוע	טוב מאד	טוב מאד	טוב מאד
Fan-In	2-8	>20	2	2-3	2-4
Fan-Out	5-10	>20	בלתי מוגבל	בלתי מוגבל	4-6
לחץ אויר	-	-	גבוה	בינוני - גבוה	נמוך
תצרוכת אויר	-	-	נמוכה	נמוכה	גבוהה
Input Interface	טוב לחיישנים	חשמליים	טוב לחיישנים	פניאומטים	
Output Interface	טוב לציוד	חשמלי	טוב לציוד	פניאומטי	דורש הגברה

Fig. 4-27 : Comparison Between Switching Elements

CHAPTER 5

HUFFMAN METODE

Feedback Memory Concept
Typical Block Diagram
Primitive Flow Diagram
Primitive Flow Table
Merge diagram
Merged Flow Table
States Assignment
Exitation and Output Functions
Ladder Circuit Realization
PLC Program
Random Inputs Systems

Huffman/PLC Combination

Huffman method is very effective in reducing the number of group relay flip-flops. It is also effective for design of systems with random inputs.

Design process by Huffman method may become very complex, from "State Assignmsnt" step, and on. Furthermore, it is based on Karnaugh maps, which eliminate the number of involved variables.

When system realization is based on PLC or on electronic gates, number of "relay" flip-flops doesn't actually effect system cost (or effect is minor)

The Huffman-PLC method :

- **implements flip-flop for each state**
- **eliminates the complexity of Huffman method,**
- **enables to implement large Karnaugh maps and Pseudo Karnaugh maps.**

Fig. 5-1 : Huffman Method - Basic

Example - A system detects cylinder piston movement direction : forward (toward A+ position), or backwords (toward A- position).

System reads limit switches a1 and a2 and outputs signals to green or red bulbs. During forward movement (input=00), green bulb must be illuminated, and during backwords movement (same input) red bulb must be illuminated,

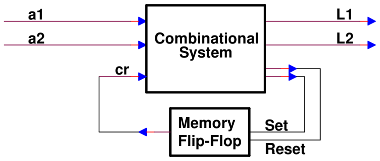
a1 a2	10	00	01	00	10	00	01	00	....
L1	0	1	0	0	0	1	0	0	......
L2	0	0	0	1	0	0	0	1	......

Note : Assume that piston changes direction only after reaching cylinder edge.

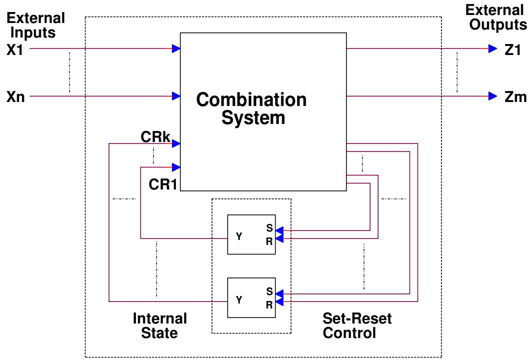
a. System Definition

b. Same Input Generates Different Outputs (Inpossible in a combinational system)

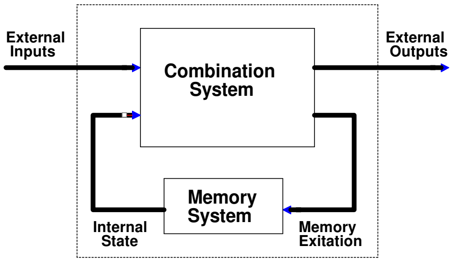
c. Adding External State Input (y) May Eliminate Problem



d. Internal Generated State Signal



f. Multiple Internal Generated State Signals



e. Sequential System General Diagram

Fig. 5-2 : Sequential System Concepts

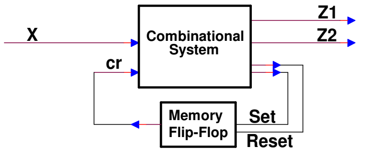
RESTRICTION : DUE TO TRANSIENT HAZARDS, ASYNCHRONOUS SYSTEM CANNOT HANDLE CHANGE OF SEVERAL INPUT SIGNALS SIMULTANEAUSLY (EXCEPT FOR SPECIFIC CASES).

a. Essential Restriction Rule

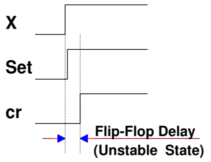
Change of an external input may couse a change of state variable, which is fed to system input. Result may cause simultaneous change of two input signals (external and internal).

In order to evoid it, simultaneous changes are replaced by sequential changes. This result is achieved due to memory (flip-flop) delay.

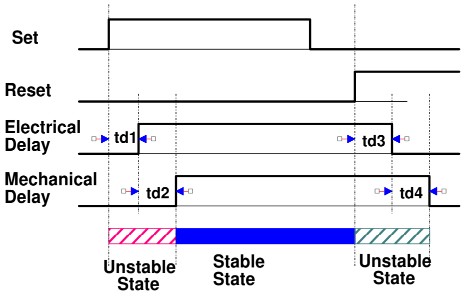
b. Hazard Problem and Its Elimination



c. Circuit Example



e. Transition Timing



Each transition from state to state is combined of unstable state followed - after delay- by stable state.

d. Relay Flip-Flop Transition Response

Fig. 5-3 : States Transition Process

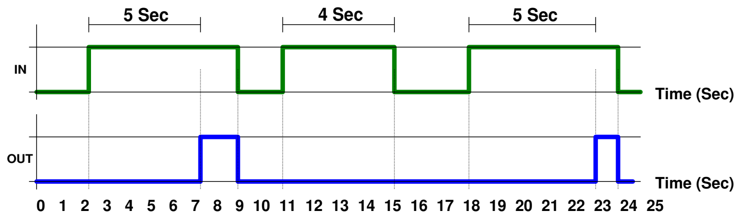
Step 1 : System requirements definition
Step 2 : Primitive flow diagram
Step 3 : Primitive flow table
Step 4 : Merge options Diagram
Step 5 : Merge groups selection
Step 6 : Merged flow table
Step 7 : States assignment
Step 8 : Output table
Step 9 : Flip-flops exitation expressions
Step 10 : Outputs expressions
Step 11 : System Logic Circuit

Fig. 5-4 : Huffman Design Steps

When input changes from "0" to "1", output is delayed T sec

When input changes from "1" to "0", output is changed immediately, and time count is reset

a. Definition



b. Example of Timing Diagram (5 Sec Delay)

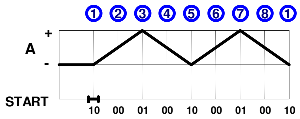
Fig. 5-5 : "On-Delay" Timer

HUFFMAN DESIGN METHOD

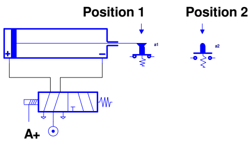
START , A+ , A- , A+ , A-

With Return Spring

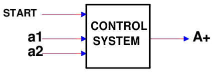
a. Sequence



b. Sequence Chart



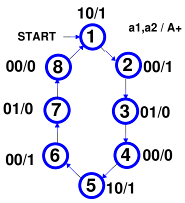
c. Cylinder Type



d. Basic Block Diagram

State	Cylinder State	Action	Output Command
1	Cylinder at A- (Start)	Start moving towards A+	A+
2	Moving towards position +	Keep moving towards A+	A+
3	Cylinder at A+	Start moving back (towards A-)	None (A-)
4	Cylinder moves towards A-	Keep moving back	None (A-)
5	Cylinder at A-	Start moving towards A+	A+
6	Cylinder moves towards A+	Keep moving towards A+	A+
7	Cylinder at A+	Start moving back (towards A-)	None (A-)
8	Cylinder moves towards A-	Keep moving back	None (A-)

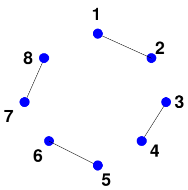
e. States Definitions



f. Primitive Flow-diagram

a1,a2	10	00	01	11
	①/1	2	-	-
	-	②/1	3	-
	-	4	③/0	-
	5	④/0	-	-
	⑤/1	6	-	-
	-	⑥/1	7	-
	-	8	⑦/0	-
	1	⑧/0	-	-

g. Primitive Flow-table



h. Merge Diagram

(refer to 5-5 for Merging rules)

{1,2) , (3,4) , (5,6) , (7,8)

i. Selected Groups

a1,a2	10	00	01	11
	①	②	3	-
	5	④	③	-
	⑤	⑥	7	-
	1	⑧	⑦	-

j. Merged Flow-table

cr \ a1,a2	10	00	01	11
cr1	①	②	3	-
cr2	5	④	③	-
cr3	⑤	⑥	7	-
cr4	1	⑧	⑦	-

k. States Assignment

cr \ a1,a2	10	00	01	11
cr1	1	1	-	-
cr2	-	0	0	-
cr3	1	1	-	-
cr4	-	0	0	-

A+

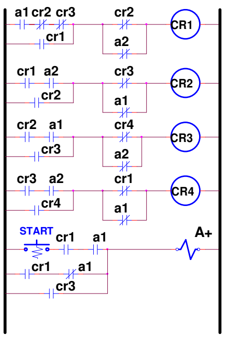
l. Output Table

~~S1 = cr4.a1~~ R1 = cr2.a2 R1' = cr2' + a2'
S2 = cr1.a2 R2 = cr3.a1 R2 '= cr3' + a1'
S3 = cr2.a1 R3 = cr4.a2 R3' = cr4' + a2'
S4 = cr3.a2 R4 = cr1.a1 R4' = cr1' + a1'
S1 = a1(cr2+cr3)' = a1.cr2'.cr3'

~~A+ = cr1~~ + cr3
A+ = START.cr1.a1 + cr1.a1' + cr3

m. Exitation and Output Functions

(refer to 5-5 for Initialization and Output Rules)



n. Relays Ladder Diagram

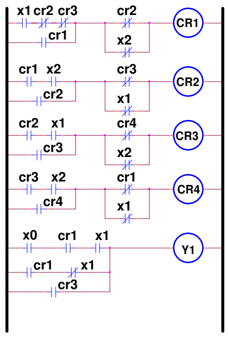
INPUTS

Start	X0
Limit contact a1	X1
Limit contact a2	X2

OUTPUTS

Solenoid A+	Y1

o. PLC I/O Tables



p. Relays Ladder Diagram

```
STORE       x1
AND NOT     cr2
AND NOT     cr3
OR          cr1
STORE NOT   cr2
OR NOT      x2
AND         STORE
OUT         CR1

AND         x2
OR          cr2
STORE NOT   cr3
OR NOT      x1
AND         STORE
OUT         CR2

AND         x1
OR          cr3
STORE NOT   cr4
OR NOT      x2
AND         STORE
OUT         CR3

AND         x2
OR          cr4
STORE NOT   cr1
OR NOT      x1
AND         STORE
OUT         CR4

STORE       x0
AND         cr1
AND         x1
STORE       cr1
AND NOT     x1
OR          STORE
OR          cr3
OUT         Y1
```

q. PLC Circuit & Program

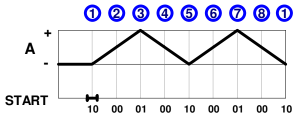
Fig. 5-6 : Sequence A+,A-,A+,A- , Huffman Method
(With Returned Springs)

HUFFMAN DESIGN METHOD

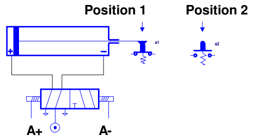
START , A+ , A- , A+ , A-

Without Return Spring

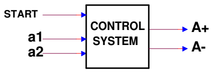
a. Sequence



b. Sequence Chart



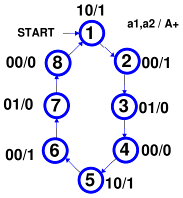
c. Cylinder Type



d. Basic Block Diagram

State	Cylinder State	Action	Output Commands	
			A+	A-
1	Position - (initial)	Start moving towards A+	1	0
2	Moving towards position +	No action required	-	0
3	A+	Start moving back (towards A-)	0	1
4	Cylinder moves towards A-	No action required	0	-
5	Cylinder at A-	Start moving towards A+	1	0
6	Cylinder moves towards A+	No action required	-	0
7	Cylinder at A+	Start moving back (towards A-)	0	1
8	Cylinder moves towards A-	No action required	0	-

e. States Definitions

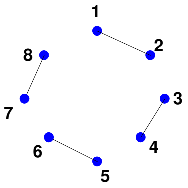


f. Primitive Flow-diagram

a1,a2

10	00	01	11	A+	A-
①	2	-	-	1	0
-	②	3	-	-	0
-	4	③	-	0	1
5	④	-	-	0	-
⑤	6	-	-	1	0
-	⑥	7	-	-	0
-	8	⑦	-	0	1
1	⑧	-	-	0	-

g. Primitive Flow-table



h. Merge Diagram

(See page 5-9 for Merging rules)

{1,2} , (3,4) , (5,6) , (7,8)

i. Selected Groups

a1,a2

10	00	01	11
①	②	3	-
5	④	③	-
⑤	⑥	7	-
1	⑧	⑦	-

j. Merged Flow-table

cr \ a1,a2	10	00	01	11
cr1	①	②	3	-
cr2	5	④	③	-
cr3	⑤	⑥	7	-
cr4	1	⑧	⑦	-

k. States Assignment

cr \ a1,a2	10	00	01	11
cr1	1	-	-	-
cr2	-	0	0	-
cr3	1	-	-	-
cr4	-	0	0	-

A+

cr \ a1,a2	10	00	01	11
cr1	00	0	-	-
cr2	-	-	1	-
cr3	0	0	-	-
cr4	-	-	1	-

A-

l. Output Tables

~~S1 = cr4.a1~~
S2 = cr1.a2
S3 = cr2.a1
S4 = cr3.a2
S1 = a1(cr2+cr3)' = a1.cr2'.cr3'

R1 = cr2.a2
R2 = cr3.a1
R3 = cr4.a2
R4 = cr1.a1

R1' = cr2' + a2'
R2 '= cr3' + a1'
R3' = cr4' + a2'
R4' = cr1' + a1'

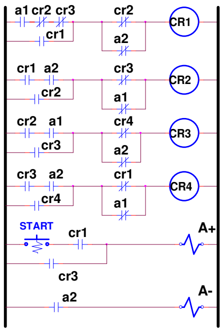
~~A+ = cr1~~ + cr3

A+ = START.cr1 + cr3

A- = a2

m. Exitation and Output Functions

(refer to 5-5 for Initialization and Output Rules)



n. Relays Ladder Diagram

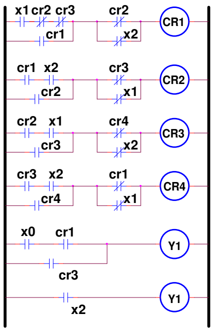
INPUTS

Start	X0
Limit contact a1	X1
Limit contact a2	X2

OUTPUTS

Solenoid A+	Y1
Solenoid A-	Y2

o. PLC I/O Tables



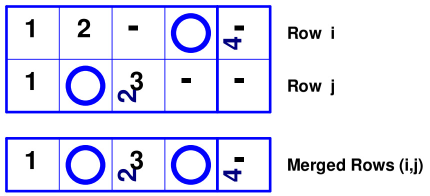
p. PLC Ladder Diagram

```
STORE      x1
AND NOT    cr2
AND NOT    cr3
OR         cr1
STORE NOT  cr2
OR NOT     x2
AND        STORE
OUT        CR1

AND        x2
OR         cr2
STORE NOT  cr3
OR NOT     x1
AND        STORE
OUT        CR2

AND        x1
OR         cr3
STORE NOT  cr4
OR NOT     x2
AND        STORE
OUT        CR3

AND        x2
OR         cr4
STORE NOT  cr1
OR NOT     x1
AND        STORE
OUT        CR4

STORE      x0
AND        cr1
OR         cr3
OUT        Y1

STORE      x2
OUT        Y2
```

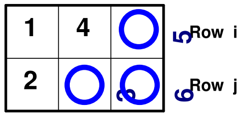
q. PLC Circuit & Program

Fig. 5-7 : Sequence A+,A-,A+,A- , Huffman Method

(Without Returned Springs)



a. Merge Available Cases



b. Merge Non-Available Cases

Fig. 5-8: Merge Flow Rows Rules

Tran	Current State	Next State	Set	Reset
0 --> 1	0	1	1	0
1 --> 0	1	0	0	1
0 --> 0	0	0	0	-
1 --> 1	1	1	-	0

a. Flip-Flop Transition Cases

Fig. 5-9: FF Exitation Rules

On system reset (or power on), all state variables are "0", meaning :

CR1=CR2=CR3=....CRi=....=CRn = 0

This state is not part of the flow table, and and therefor is not taking care of. As a result, if we refere only to the flow table, we can't force the initial state CR to become "1".

Most (or all) set/reset functions are product of state-variable (or more), but since all of them are 0, all functions are inhibited.

Therefor, driving the system to its expected initial state, must be done by a function that doesn't depend on any operated CRi, but only on non-operated CRi' .

This is acheived by writing the initial function in its negative conditions, and negate it (similar to implementing the "0" Karnaught table).

So, instead of writing the CR states where the transition to that state must be carried on, we write the states where the transition is forbidden, as a function of CR or several CRs, and then negate it.

Negating this expression, using De-Morgan rules, bring to a function that depends on CR', and this may be carried on.

System Inialization

In most cases, only steady-state output is specified.

But transition between steady states is done by pathing through a non-steady state : transients. If we don't take care of transition states, say define it as don't care, system output may be wrong during those transients.

Usually, transient time is very short with respect to steady states duration.

In some cases, we may ignore the need to define transitions output, usually when output inertion is high. For example, standard 220 valve is actuated by AC current, meaning that during each cycle its current is 0 twice, but it illuminates as required. This a simple case where due to high enersion, transient outputs do not affect the expected output.

Generally, if there is not enough information about the importance of transient outputs, the following rules are used :

Rule 1 : If outputs of 2 steady states are same, the output of the transition between them must be equal to the steady-state output.

Rule 2 : If outputs of 2 steady states are different from each other, the output of the transition between them may be defined as don't care.

Transient Outputs

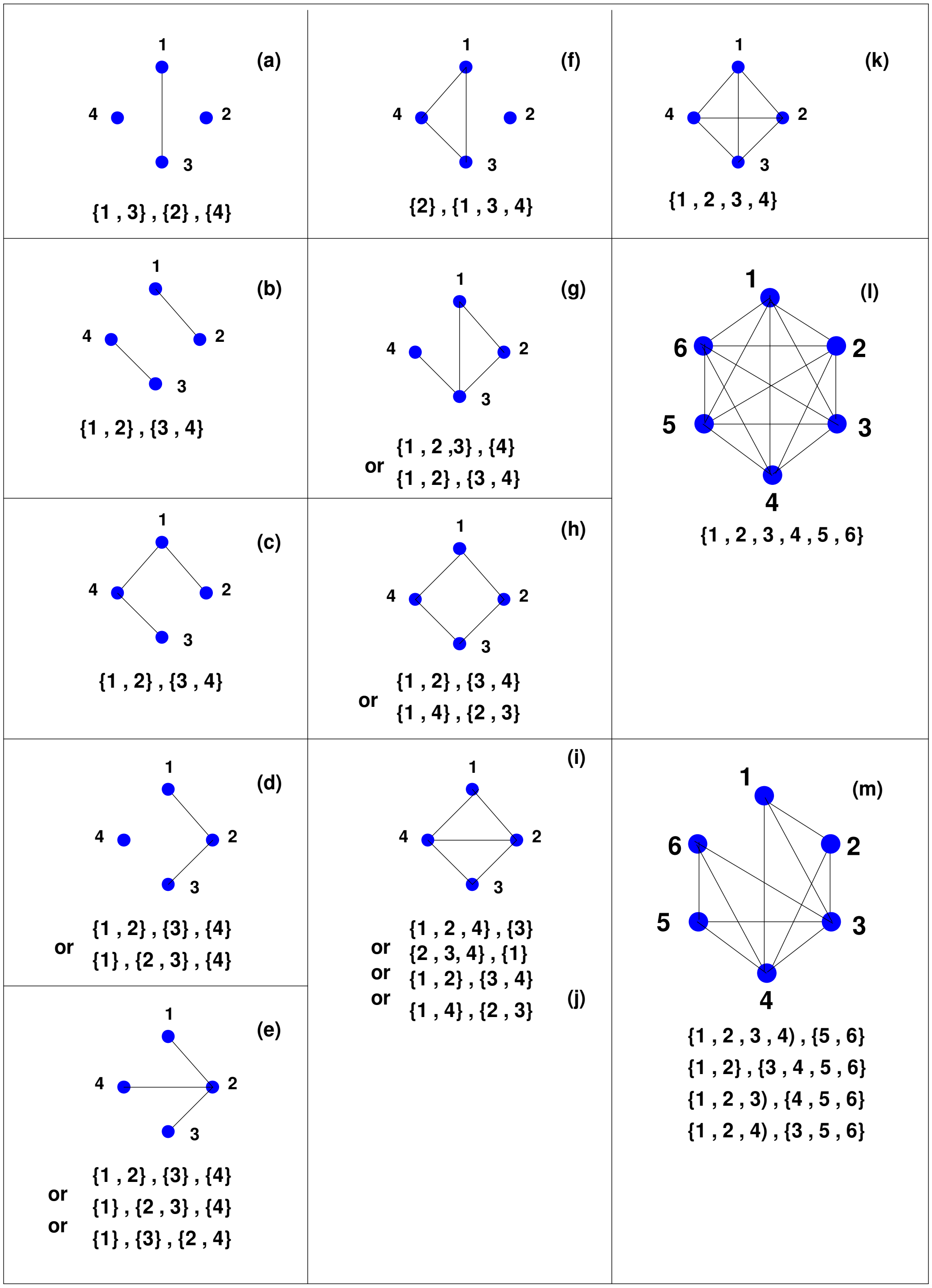
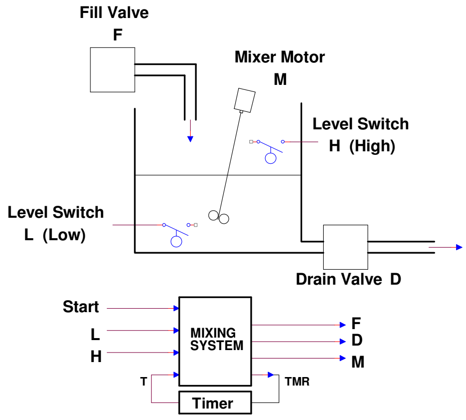


Fig. 5-10 : States Grouping Examples

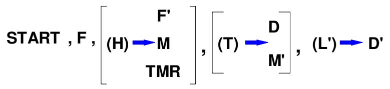
A mixing system requires filling a tank with liquid, mix it for a specific time period and then drain the liquid out.

An automatic process is represented as follows :

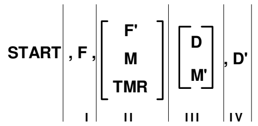
1. Sequence starts by pressing START buttom. This opens FILL valve.
2. Liquid fills tank until HIGH level sensor is operated.
3. At that time FILL valve is closed, mixing motor is turned on, and a timer is activated
4. On timeout, motor is turned off, and DRAIN valve is opened.
5. When tank is empty, LOW level sensor contacts open, and DRAIN valve is closed.



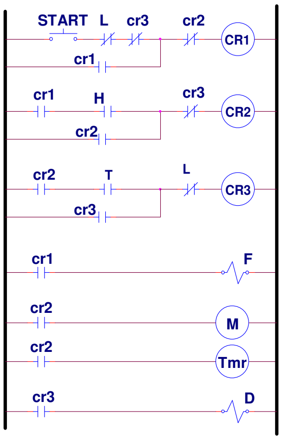
a. Process presentation and requirements definitions



b. Process Sequence



c. Groups Partition



d. Cascade Contacts Circuit

Fig. 5-11 : Automatic Mixing System

Cascade Method

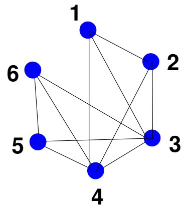
5-7

Process is same as defined ai Fig. 6-25/a .

a. Process definitions

LHT 000	100	110	111	101	F	D	M	Ymr
①	2	-	-	-	1	0	0	0
-	②	3	-	-	1	0	0	0
-	-	③	4	-	0	0	1	1
-	-	-	④	5	0	1	-	1
-	6	-	-	⑤	0	1	0	0
1	⑥	-	-	-	0	1	0	0

b. Primitive Flow Table



c. Merge Diagram

{1 , 2 , 3} , {4 , 5 , 6}
{1 , 2 , 3 , 4} , {5 , 6}
{1 , 2} , {3 , 4 , 5 , 6}

d. Minimized Groups Partition

{1 , 2 , 3} , {4 , 5 , 6}

e. Selected Groups Partition

CR \ LHT	000	100	110	111	101
1	①	②	③	4	-
0	1	⑥	-	④	⑤

f. Merged Flow Table

cr1,T \ L.H	00	10	11	01
00	1	⑥	-	-
01	-	⑤	④	-
11	-	-	4	-
10	①	②	③	-

g. "Path" Map

cr1,T \ L.H	00	10	11	01
00	10	0-	--	--
01	--	0-	0-	--
11	--	--	01	--
10	-0	-0	-0	--

h. Exitation Map (SR)

START

cr1,T \ L.H	00	10	11	01
00	0	0	-	-
01	-	0	0	-
11	-	-	(0)	-
10	1	1	0	-

F

START

cr1,T \ L.H	00	10	11	01
00	0	1	-	-
01	-	1	1	-
11	-	-	-	-
10	0	0	0	-

D

cr1,T \ L.H	00	10	11	01
00	(0)	0	-	-
01	-	0	-	-
11	-	-	-	-
10	0	0	1	-

M

cr1,T \ L.H	00	10	11	01
00	(0)	0	-	-
01	-	0	1	-
11	-	-	(1)	-
10	0	0	1	-

TMR

i. Output Tables

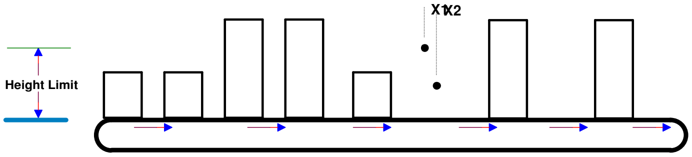
S = L'.START
F = H'.cr1
M = H
R = tmr
D = L.cr1'
TMR = H

j. System Functions

Fig. 5-12 : Automatic Mixing System

Huffman Method

1. Different size elements are transferred on a conveyor belt.
2. All elements that are below specific height limit must be rejected.
3. Reject is performed by opening a reject door - Z=1.
4. Photocells X1 and X2 are placed in specific locations.
5. X2 is covered by all sizes. X1 is covered by elements over the defined specific heigh limit.

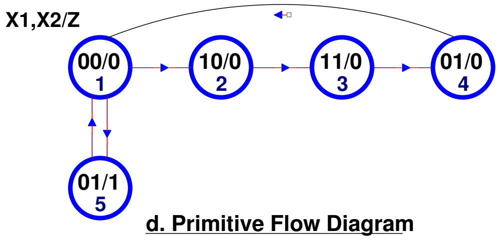


a. System Definition

b. System Block Diagram

State 1 : Check area empty (00)
State 2 : High element enters area (10)
State 3 : High element in check area (11)
State 4 : High element starts leaving area (01)
State 5 : Low element in inside check area (01). Pushed out

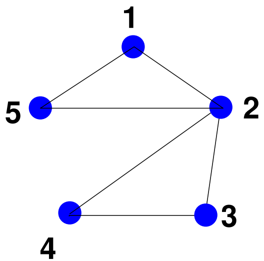
c. States Definitions



d. Primitive Flow Diagram

X1,X2 00	10	11	01	Z
①	2	-	5	0
-	②	3	-	0
-	-	③	4	0
1	-	-	④	0
1	-	-	⑤	1

e. Primitive Flow Table



f. Merge Diagram

{1,2,5},{3,4}

{1,5},{2,3,4}

g. Merge Options

{1,5},{2,3,4}

h. Selected Options

X1,X2 00	10	11	01
①	2	-	⑤
1	②	③	④

i. Merged Flow Table

cr1 \ X1,X2	00	10	11	01
0	①	2	-	⑤
1	1	②	③	④

j. State Assignment

cr1 \ X1,X2	00	10	11	01
0	0	(0)	-	1
1	(0)	0	0	0

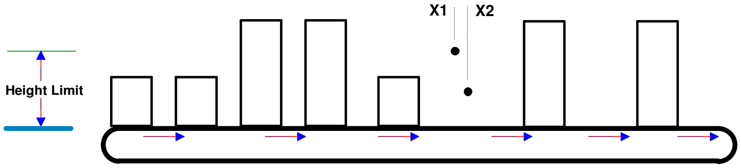
k. Output Table

S1 = X1 (S1 = X1.X2')
R1 = X1'.X2' (R1 = X1'.X2'.cr1)
Z = X2.cr1' (Z = X1'.X2.cr1')

l. System Functions

Fig. 5-13 : Height Detector System Design

1. Different size elements are transferred on a conveyor belt.
2. All elements that are below specific height limit must be rejected.
3. Reject is performed by opening a reject door - Z=1.
4. Photocells X1 and X2 are placed in specific locations.
5. X2 is covered by all sizes. X1 is covered by elements over the defined specific heigh limit.



INPUTS

Photo-cell X1	X1
Photo-cell X2	X2

OUTPUTS

Cylinder Z	Y1

d. PLC I/O List

$S1 = X1$ $(S1 = X1.X2')$

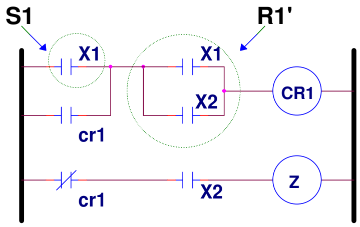
$R1 = X1'.X2'$ $(R1 = X1'.X2'.cr1)$

$Z = X2.cr1'$ $(Z = X1'.X2.cr1')$

a. Summary of Fig. 5-13

```
STORE      x1
OR         cr1
STORE      x1
OR         x2
AND        STORE
OUT        CR1
.....................
STORE NOT  cr1
AND        x2
OUT        Y1
```

e. PLC Program

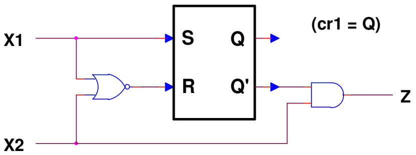


b. Relays Implementation

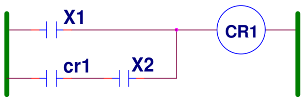
$CR1 = (X1 + cr1).(X1 + X2) =$

$= X1.X1 + X1.X2 + cr1.X1 + cr1.X2 =$

$= X1+ cr1.X2$



c. Gates Implementation



f. cr1 Function Simplification

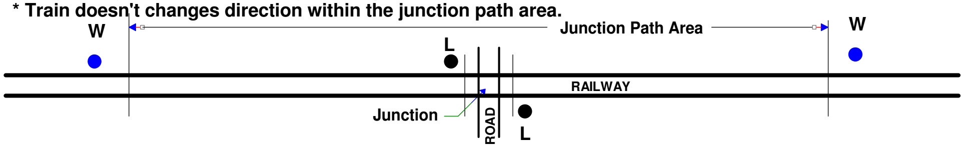
Fig. 5-14 : Realiation of Fig. 5-13

A red light signal is placed in the junction of road and raillway. Its task is warning cars not to cross the junction. Another red light signal is placed far from the junction. Its task is warning other trains not to enter junction area.

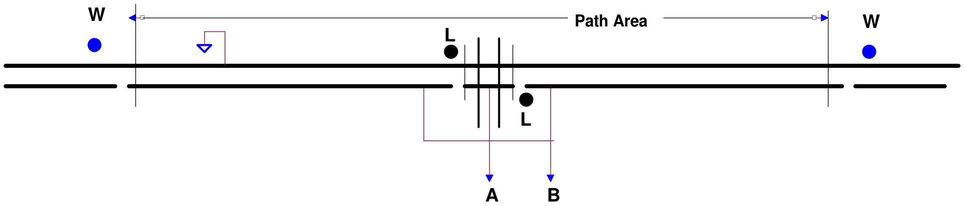
Train must operate red light as soon as it enters "junction path area", far from the junction, and turn it off as soon as last waggon leaves the junction, near junction.

Trains may go in two direction, but the following assumption are known :

- * Only a single train crosses the junction path area, at a time.
- * Train doesn't changes direction within the junction path area.

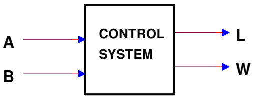


One of the tracks is connected to common node (Ground), while the other track is splitted - within the junction path area - into 3 sections, that are physically joined, but electrically isolated from the graounded track, and from tracks outside junction path area.
Mid-section is also isolated from the other two, as seen in the drawing.



When the junction path area is empty (no trains inside) all 3 sections are disconnected from common track, thus providing "0" signal on each section.
When the train crosses a track section, it connects it electrically to the common track, thus changing its signal into "1".
Signal returns to "0" as soon as last waggon leaves the appropriate track section.
Signaling control system senses A and B signals, and turns red lights on or off, accordingly.

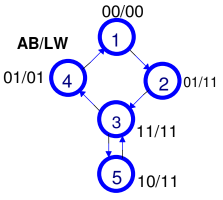
a. System description



b. Basic Block Diagram

State	Position	AB	LW
1	Junction area empty	00	00
2	Train enters area	01	11
3	Train over central area	11	11
4	Train leaving central area	01	01
5	"Short" train on central area	10	11

c. States Definitions

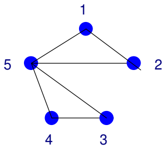


d. Flow-diagram (primitive)

AB

00	01	11	10	L	W
①	2	-	-	0	0
-	②	3	-	1	1
-	4	③	5	1	1
1	④	-	-	0	1
-	-	3	⑤	1	1

e. Primitive Flow-table



f. Merge Diagram

{1,2,5} , (3,4)
{3.,4,5} , (1,2)

g. Merge Options

{1,2,5} , (3,4)

h. Selected Groups

AB

00	01	11	10
①	②	3	⑤
1	④	③	5

i. Merged Flow-table

AB

cr1	00	01	11	10
0	①	②	3	⑤
1	1	④	③	5

j. States Assignment

AB

cr1	00	01	11	10
0	00	11	(11)	11
1	(0-)	01	11	(11)

LW

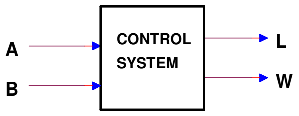
k. Output Table

S1 = AB
R1 = B'   (R1' = B)

L = A + cr1'.B
W = A + B

l. System Functions

Fig. 5-15 : Traffic-Light Control System Design

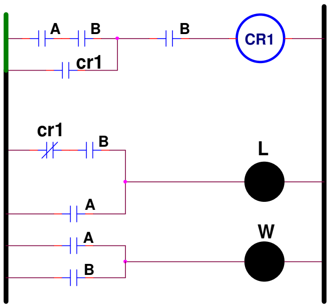


a. Basic Block Diagram

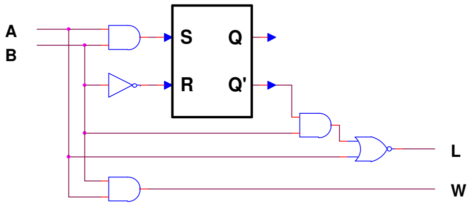
S1 = AB
R1 = B' (R1' = B)

L = A + cr1'.B
W = A + B

b. System Functions



c. Realys SYstem Realization

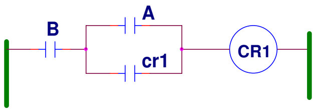


d. Gates Implementation

CR1 = (A.B + cr1)B =

=A.B.B + cr1.B =

=A.B + cr1.B = B(a + cr1)



h. CR1 Function Simplification

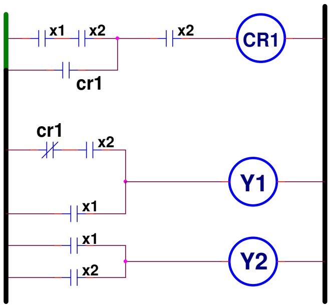
INPUTS

A B	X1 X2

OUTPUTS

Light L	Y1
Warning W	Y2

e. PLC I/O Tables



f. PLC Circuit

```
STORE       x1
AND         x2
OR          cr1
AND         x2
OUT         CR1

STORE NOT   cr1
AND         x2
OR          x1
OUT         Y1

STORE       x1
OR          x2
OUT         Y2
```

g. PLC Program

Fig. 5-16 : Realization of Fig. 5-15

1. System FAIL status is exspressed by an external signal F=1.
   It must activate siren S, until operator confirmation
2. Operator confirms by pressing C buttom. This de-activates siren.
3. If FAIL signal still exists, confirmation turns on flash light L =1.
4. After confirmation, flash is turned off a soon as fail terminates.

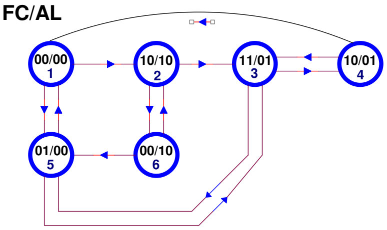
a. System Definition

b. System Block Diagram

(0) -

State	Position	Error Status	Key Status	Alarm Status	Warning Status	FC AL
1	Idle State	NO	OFF	NO	NO	00 00
2	Error exists. Not yet confirmed	YES	OFF	YES	NO	10 10
3	Error exists and is being confirmed (Error exists and operator presses key)	YES	ON	NO	YES	11 01
4	Error exists and been confirmed (Error exists and operator releases key)	YES	OFF	NO	YES	11 01
5	Confirmed key depressed while no error occurs	NO	ON	NO	NO	01 00
6	Error terminated but not yet confirmed (Error exists and operator releases key)	NO	OFF	YES	NO	00 10

c. States Definitions



d. Primitive Flow Diagram

{3,4,5} {2.6} {1}

{1,5} {2.6} {3,4}

g. Merge Options

{1,5} {2.6} {3,4}

h. Selected Options

FC	00	10	11	01
cr1	①	2	3	⑤
cr2	⑥	②	3	5
cr3	1	④	③	5

i. Merged Flow Table & States Assignment

FC	00	10	11	01	S	L
	①	2	-	5	0	0
	6	②	3	-	1	0
	-	4	③	5	0	1
	1	④	3	-	0	1
	1	-	3	⑤	0	0
	⑥	2	-	5	1	0

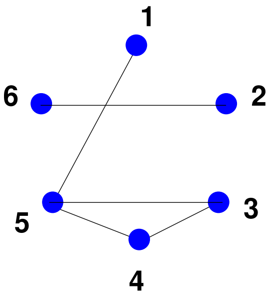
e. Primitive Flow Table

FC	00	10	11	01
cr1	0	-	(0)	0
cr2	1	1	-	-
cr3	(0)	0	0	(0)

j. Output Table of S

FC	00	10	11	01
cr1	0	(0)	-	0
cr2	0	0	-	(0)
cr3	-	1	1	-

k. Output Table of L



f. Merge Diagram

S1 = F'.C'.cr2' + F'.C = F'(C + cr2')

S2 = cr1.F.C'

S3 = F.C

R1 = cr2.F.C' + F.C = F(C + Ccr2)

R2 = F'.C + F.C = C

R3 = cr1.F'.C' + cr1.F'.C = cr1.F'   (*)

S = cr2

L = cr3

(*) Non-minimal cell selction enables function minimaztion

l. Exitation & Output Functions

Fig. 5-17: Alarm System Design

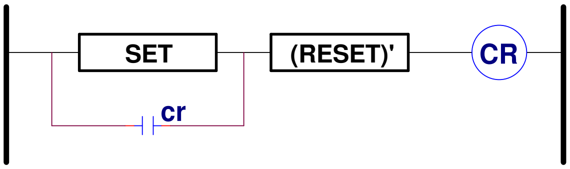
S1 = F'C'.cr2' + F'.C = F'(C + cr2')
S2 = cr1.F.C'
S3 = F.C
R1 = cr2.F.C' + F.C = F(C + cr2)
R1' = F'+(C'.cr2')

R2 = F'.C + F.C = C
R2' = C'

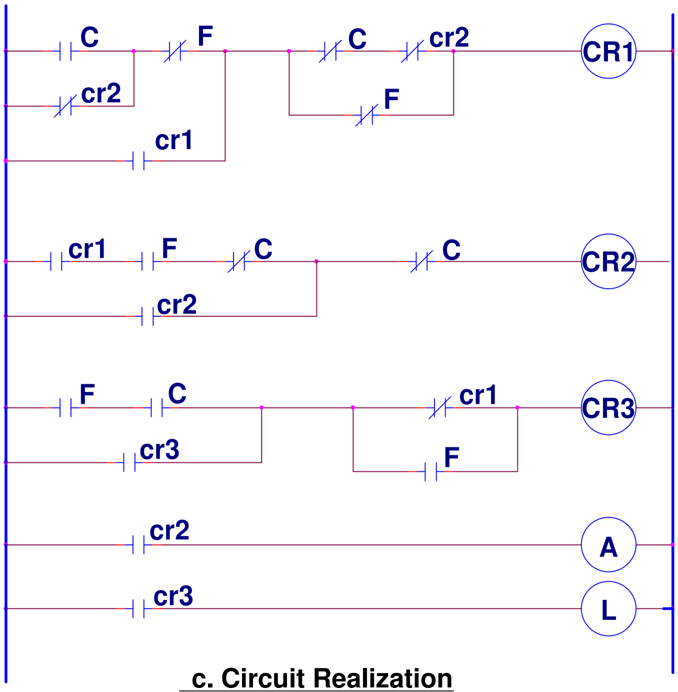
R3 = cr1.F'.C' + cr1.F'.C = cr1.F'
R3' = cr1'+F

A = cr2 L = cr3

a. Exitation & Output Functions



b. Typical Relay Flip-Flop



c. Circuit Realization

Fig. 5-18: Realization of Fig. 5-17 by Relays System

$S1 = F'C'.cr2' + F'.C = F'(C + cr2')$
$S2 = cr1.F.C'$
$S3 = F.C$
$R1 = cr2.F.C' + F.C = F(C + cr\ 2)$
$R1' = F'+(C'.cr2')$

$R2 = F'.C + F.C = C$
$R2' = C'$

$R3 = cr1.F'.C' + c\underline{r1.F'.}C = cr1.F'$
$R3' = cr1'+ F$

$A = cr2$ $L = cr3$

a. Exitation & Output Functions

INPUTS

F C	X1 X2

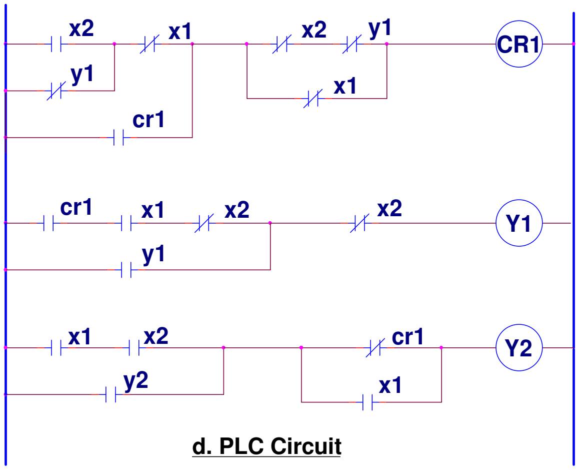
OUTPUTS

Alarm A	Y1
Flash L	Y2

b. PLC I/O Tables

Y1 → CR2
Y2 → CR3

c. CR Equivalence

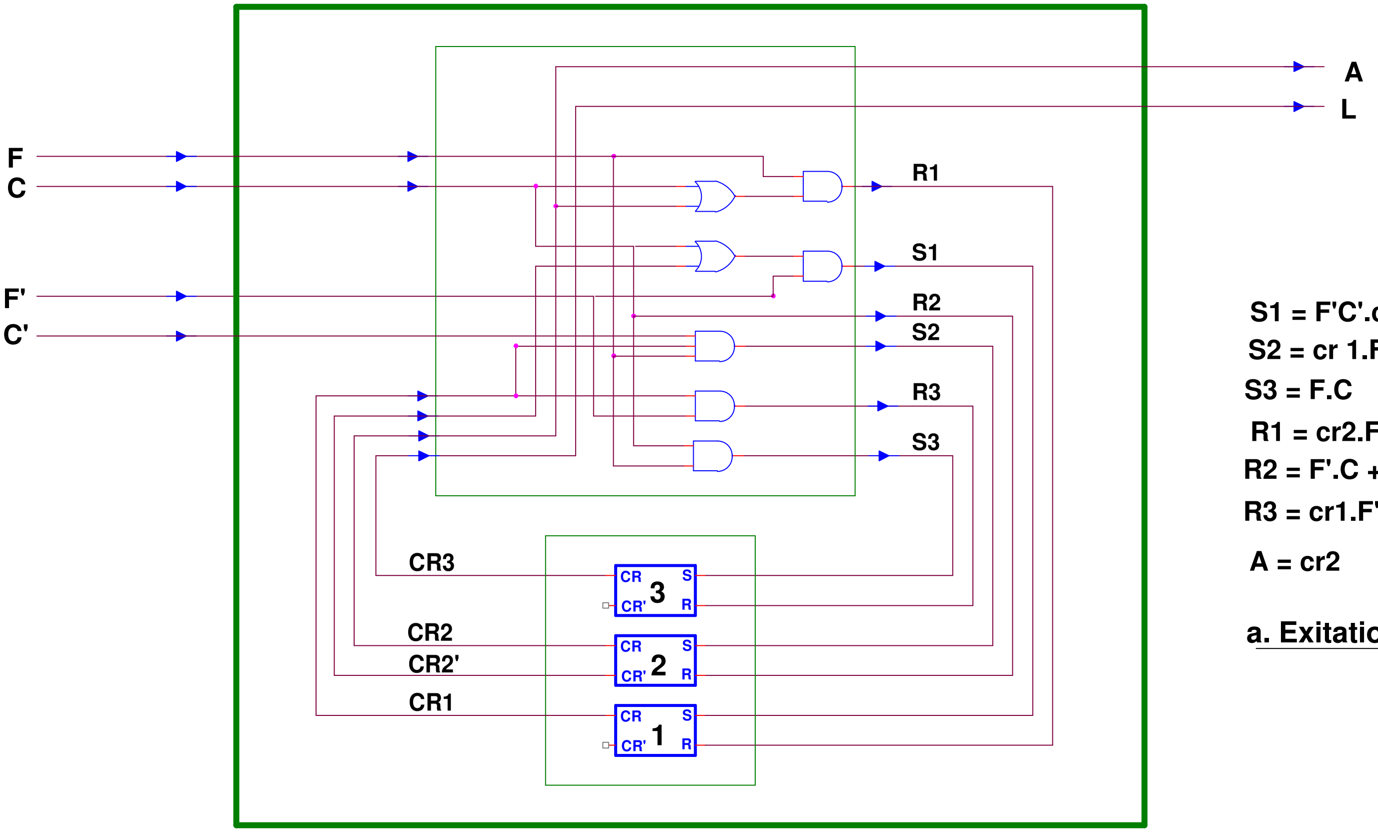


d. PLC Circuit

```
STORE        x2
OR NOT       y1
AND NOT      x1
OR           cr1
STORE NOT    x2
AND NOT      y1
OR NOT       x1
AND          STORE
OUT          CR1

AND          x1
AND NOT      x2
OR           y1
AND NOT      x2
OUT          Y1

STORE        x1
AND          x2
OR           y2
STORE NOT    cr1
OR           x1
AND          STORE
OUT          Y2
```

e. PLC Program

Fig. 5-19: Realization of Fig. 5-17 by PLC



b. Circuit Realization

S1 = F'C'.cr2' + F'.C = F'(C + cr2')

S2 = cr 1.F.C'

S3 = F.C

R1 = cr2.F.C' + F.C = F(C + cr2)

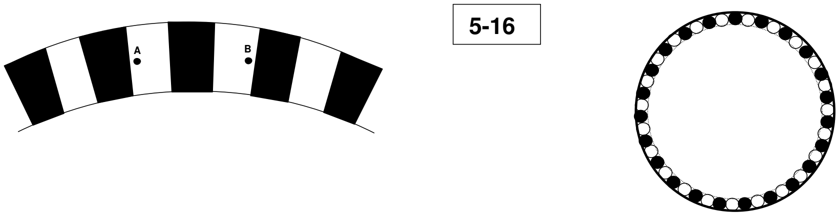
R2 = F'.C + F.C = C

R3 = cr1.F'.C' + cr1.F'.C = cr1.F'

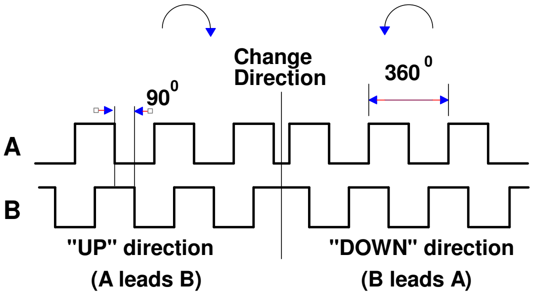
A = cr2 L = cr3

a. Exitation & Output Functions

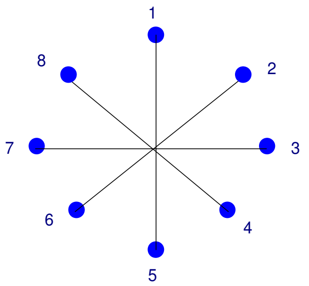
Fig. 5-20: Realization of Fig. 5-17 by Electronic Gates System



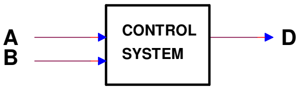
a. Disk Partitions ans Sensors Arrangement



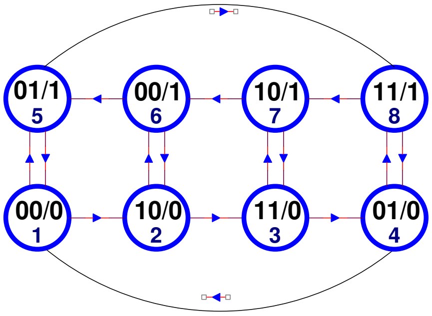
b. Encoder Code Waveforms for Both Directions



f. Merge Diagram



c. Contro,l System Block Diagram



d. Primitive Flow-diagram

AB	00	01	11	10
cr1	①	⑤	8	2
cr2	⑥	5	3	②
cr3	6	4	③	⑦
cr4	1	④	⑧	7

g. Merged Flow-table & State Assignment

AB	00	01	11	10
cr1	O	1	(1)	(0)
cr2	1	(1)	(0)	O
cr3	(1)	(0)	O	1
cr4	(0)	O	1	(1)

D

h. Output Table

AB 00	01	11	10	D
①	5	-	2	O
6	-	3	②	O
-	4	③	7	O
1	④	8	-	O
1	⑤	8	-	1
⑥	5	-	2	1
6	-	3	⑦	1
-	4	⑧	7	1

e. Primitive Flow-table

(S1 = cr2.A'B + cr4.A'B')
S2 = cr1.AB' + cr3.A.'B'
S3 = cr2.A.B + cr4.A.B'
S4 = cr1.A.B + cr3.A'B

R1 = cr2.A.B' + cr4.A.B
R2 = cr 1.A'B + cr3.AB
R3 = cr2.A'.B' + cr4.A'B
R4 = cr 1.A'.B' + cr3.AB'

D = cr1.B + cr2.A' + cr3.B' + cr4.A

i. Exitation and Output Functions

Initially, while A=B=0, and all Flip-flops are in reset state (CRi=0).

At that state, Flip-flop 1 must be set, with no confilict later, that is to say, while CR2 and CR3 are not set.

S1 = cr2.A'B + cr2'.cr3'.A'B'

j. Corrected System Initialization

Fig. 5-21 : Motor Direction Detection System Design

$S1 = A'(cr2.B + cr2'.cr3'.B')$

$S2 = B'(cr1.A + cr3.A')$

$S3 = A(cr2.B + cr4.B')$

$S4 = B(cr1.A + cr3.A')$

$R1 = A(cr2.B' + cr4.B)$

$R2 = B(cr1.A' + cr3.A)$

$R3 = A'(cr2.B' + cr4.B)$

$R4 = B'(cr1.A' + cr3.A)$

$R1' = A' + (cr2' + B).(cr4' + B')$

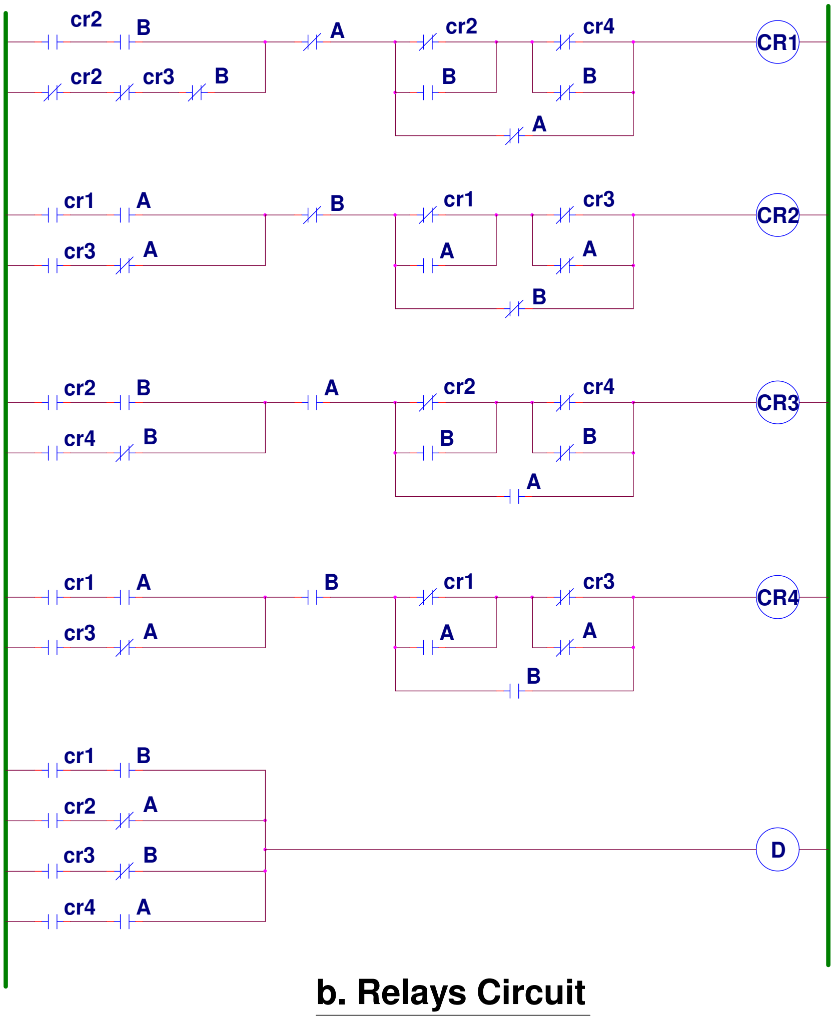
$R2' = B' + (cr1' + A).(cr3' + A')$

$R3' = A + (cr2' + B).(cr4' + B')$

$R4' = B + (cr1' + A).(cr3' + A')$

$D = cr1.B + cr2.A' + cr3.B' + cr4.A$

a. Exitation and Output Functions



b. Relays Circuit

Fig. 5-22: Realization of Fig. 5-21 by Relay System

$S1 = A'(cr2B + cr2'.cr3'.B')$

$S2 = B'(cr1.A + cr3.A.')$

$S3 = A(cr2.B + cr4.B')$

$S4 = B(cr1.A +cr3.A')$

$R1 = A(cr2.B' + cr4.B)$

$R2 = B(cr1.A' + cr3.A)$

$R3 = A'(cr2.B' + cr4.B)$

$R4 = B'(cr1.A' + cr3.A)$

$R1' = A' + (cr2' + B).(cr4' + B')$

$R2' = B' + (cr1' + A).(cr3' + A')$

$R3' = A + (cr2' + B).(cr4' + B')$

$R4' = B + (cr1' + A).(cr3' + A')$

$D = cr1.B + cr2.A' + cr3.B' + cr4.A$

a. Exitation and Output Functions

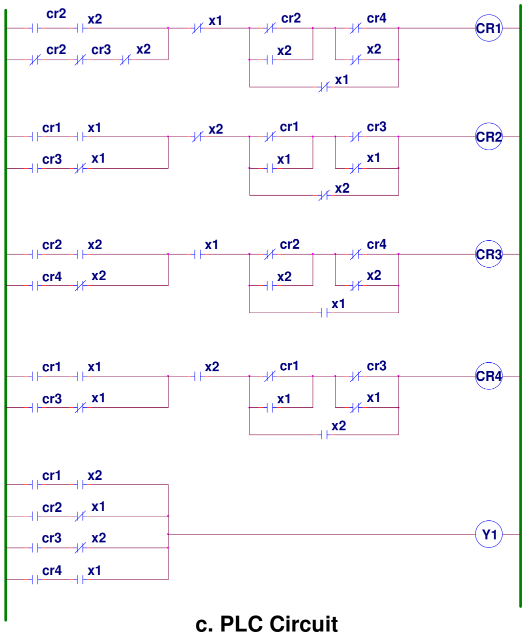
INPUTS

A B	X1 X2

OUTPUTS

Direction D	Y1

b. PLC I/O Tables



c. PLC Circuit

```
STORE       cr2
AND         x2
STORE NOT   cr2
AND NOT     cr3
AND NOT     x2
OR          STORE
AND NOT     x1
STORE NOT   cr2
OR          x2
STORE NOT   cr4
OR NOT      x2
AND         STORE
OR NOT      x1
AND         STORE
OUT         CR1
- - - - - - - - - - -
AND         x1
STORE       cr3
AND NOT     x1
OR          STORE
AND NOT     x2
STORE NOT   cr1
OR          x1
STORE NOT   cr3
OR NOT      x1
AND         STORE
OR NOT      x2
AND         STORE
OUT         CR2
- - - - - - - - - - -
AND         x2
STORE       cr4
AND NOT     x2
OR          STORE
AND NOT     x1
STORE NOT   cr2
OR          x2
STORE NOT   cr4
OR NOT      x2
AND         STORE
OR NOT      x1
AND         STORE
OUT         CR3
- - - - - - - - - - -
STORE       cr1
AND         x1
STORE       cr3
AND NOT     x1
OR          STORE
AND         x2
STORE NOT   cr1
OR          x1
STORE NOT   cr3
OR NOT      x1
AND         STORE
OR NOT      x2
AND         STORE
OUT         CR4
- - - - - - - - - - -
STORE       cr1
AND         x2
STORE       cr2
AND NOT     x1
OR          STORE
STORE       cr3
AND NOT     x2
OR          STORE
STORE       cr4
AND         x1
OR          STORE
OUT         Y1
```

d. PLC Program

Fig. 5-23: Realization of Fig. 5-21 by PLC

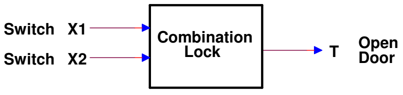
1. A security door is controlled by two input switches : X1 and X2 .
2. The door is opened and closed, by entering the following sequence :

X1,X2 = 00 , 10 , 11 , 10 , 11 00

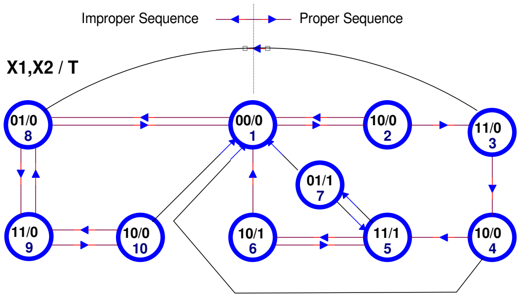
Open (T=1) Close (T=0)

Actually, door is opened when latest 5 inputs satisfy the sequence 00 , 10 , 11 , 10 , 11 , and closed as soon as both inputs become 0 (X1X2=00).

a. System Definition



b. System Block Diagram

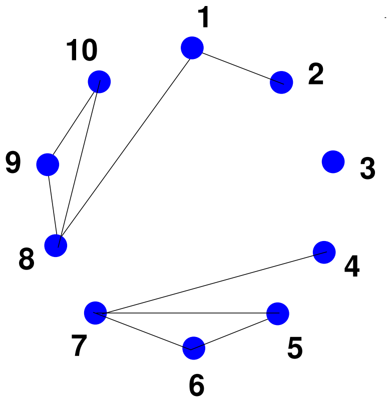


c. Primitive Flow Diagram

X1,X2

00	01	11	10	T
①	8	-	2	0
1	-	3	②	0
-	8	③	4	0
1	-	5	④	0
-	7	⑤	6	1
1	-	5	⑥	1
1	⑦	5	-	1
1	⑧	9	-	0
-	8	⑨	10	0
1	-	9	⑩	0

d. Primitive Flow Table



e. Merge Diagram

{1,2},{3},{4},{5,6,7},{8,9,10}

f. Selected Merge Partitions

X1,X2

	00	01	11	10
cr1	①	8	3	②
cr2	-	8	③	4
cr3	1	-	5	④
cr4	1	⑦	⑤	⑥
cr5	1	⑧	⑨	⑩

g. Merged Flow Table & States Assignment

X1,X2

	00	01	11	10
cr1	0	-	-	0
cr2	-	-	0	-
cr3	-	-	-	0
cr4	-	1	1	1
cr5	-	0	0	0

h. Output Table

S1 = X1'.X2'
S2 = cr1.X1.X2
S3 = cr2.X1.X2'
S4 = cr3.X2
S5 = cr4'.X1'.X2

R1 = cr2.X1.X2 + cr5.X1'.X2
R2 = cr3.X1.X2' + cr5.X1'.X2
R3 = cr4.X1.X2 +X1'.X2'
R4 = X1'.X2'
R5 = X1'.X2'

T = cr4

i. System Functions

Fig. 5-24 : Combination Lock System Design

CHAPTER 6

PNEUMATIC CONTROL SYSTEMS

Pneumatic Cascade
Efficient Valve Implementation
Emergency-Stop Modes

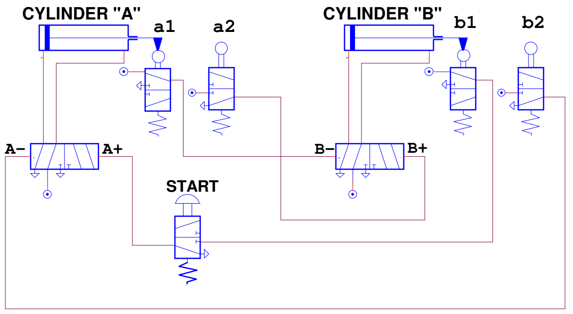
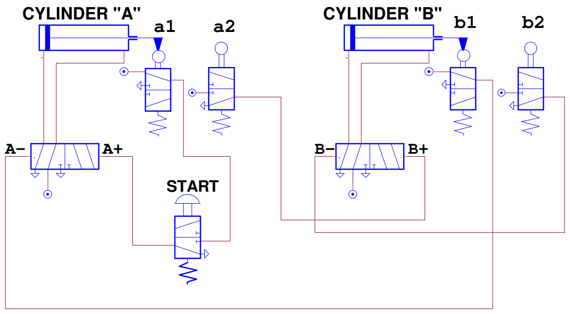
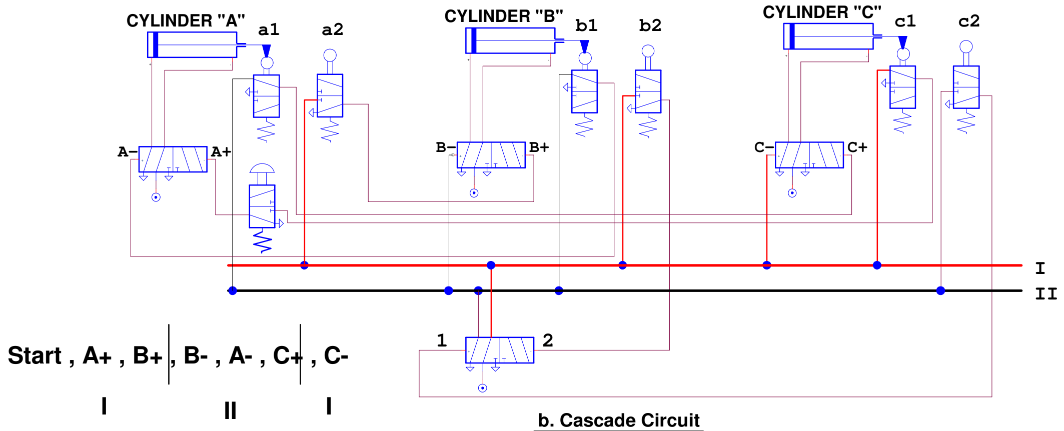


Fig. 6-1 : Intuitive Design for Sequence Start , A+,B+,A-,B-



START (A+) Conflicts With A- Actuation (by b1)
More Conflicting States

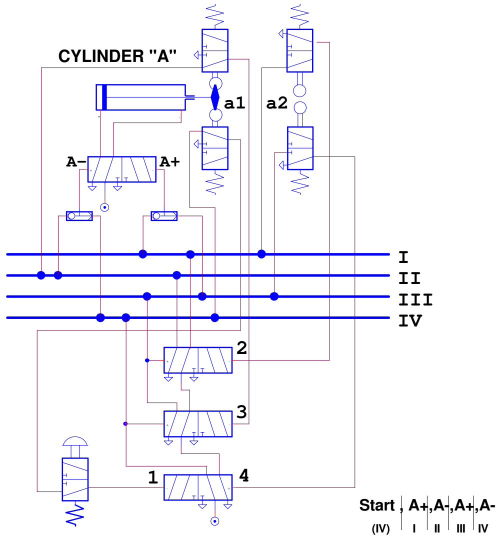
Fig. 6-2 : Sequence Start , A+,B+,B-,A-



a. Cascade Groups Partitioning

b. Cascade Circuit

Fig. 6-3 : Cascade Design for Sequence Start , A+,B+,B-,A-,C+,C-



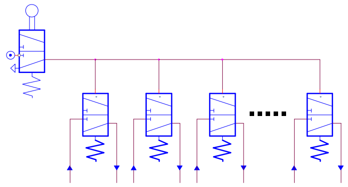


Fig. 6-5: Multiple Limit Valves Replacement

Fig. 6-4 : Cascade System for Sequence Start , A+,A-,A+,A- (Cascade)

Pure pneumatic control system

Simplicity and realiability

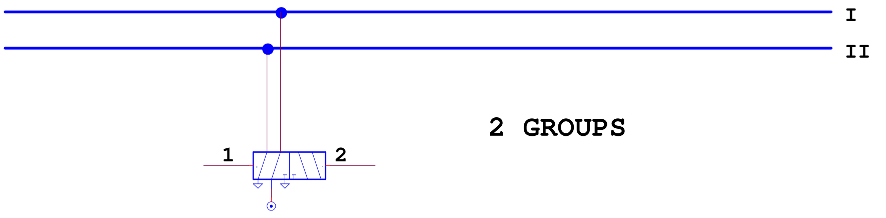
Limited to control cylinders with actuation valves that satisfy the following restrictions :

- **Valves do not have return springs**
- **Valves are operated pneumatically (no solenoids)**

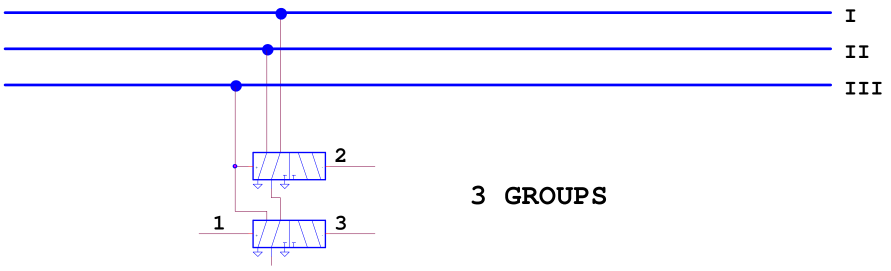
Group Partitioning differences between pneumatic and relays cascade are :

- **There is always an active group, even when not operating (START is also included within a group)**
- **Last Group may be merged with first group, provided that groups do not conflict**

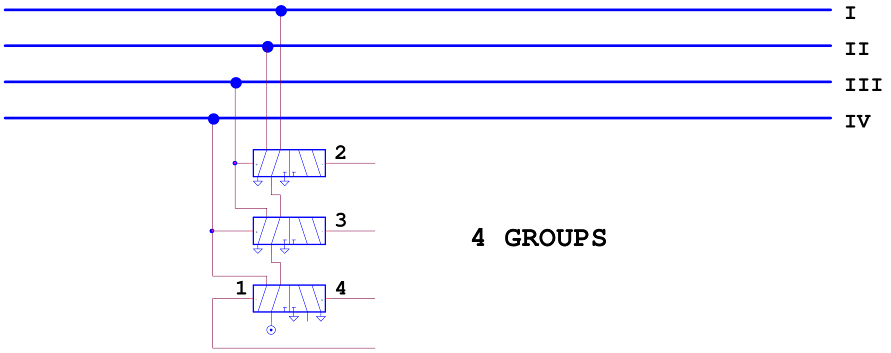
Fig. 6-6 : Pneumatic Cascade Basic Concept



a. 2-Groups Manifold Lines Configuration

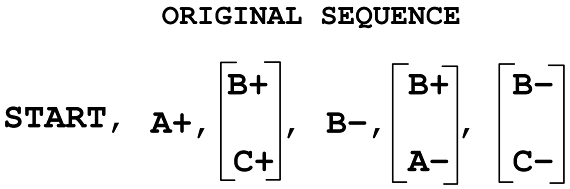


b. 3-Groups Manifold Lines Configuration

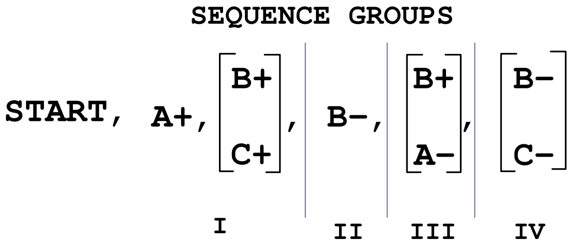


c. 4-Groups Manifold Lines Configuration

Fig. 6-7 : Pneumatic Cascade Group Configurations



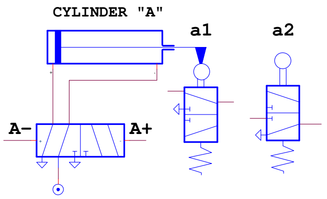
a. Process Sequence Definition



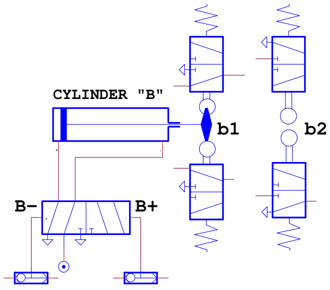
b. Sequence Groups Partioning

1. cylinder actuated by valves without return springs.
2. A set of two limit valves.

1. cylinder actuated by valves without return springs
2. Two sets of two limit valves.
3. A two-input OR gate for each valve control.

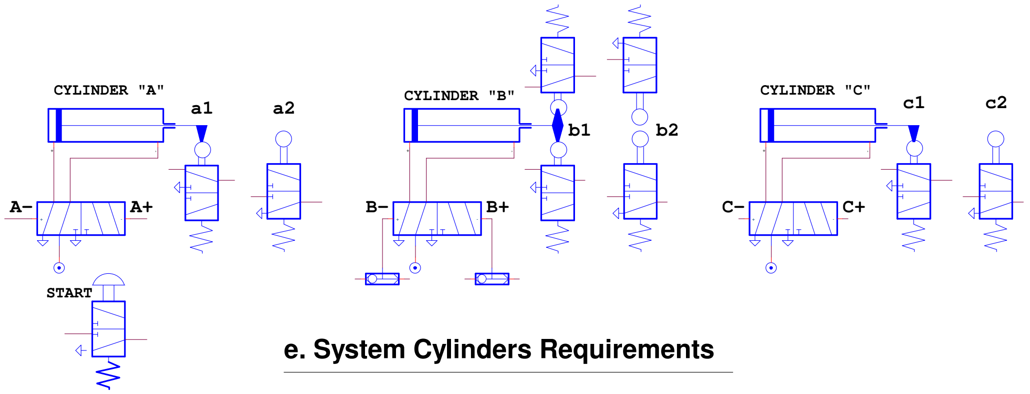


c. Single-Cycle Cylinder Requirement



d. Two-Cycle Cylinder Requirement

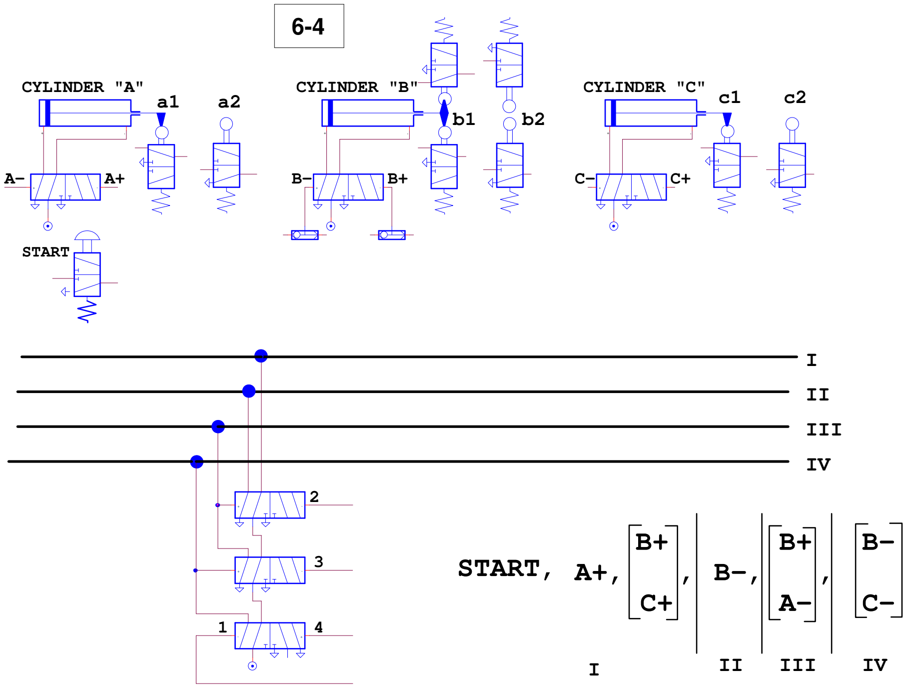
1. System consists of 3 cylinders, each equipped with limit-valves
2. Cylinders "A" and "C" perform single cycle (each), and require a single set of limit valves.
3. Cylinder "B" performs two cycles, and requires two sets of limit-valves, and OR valves.
4. A START valve is also required.



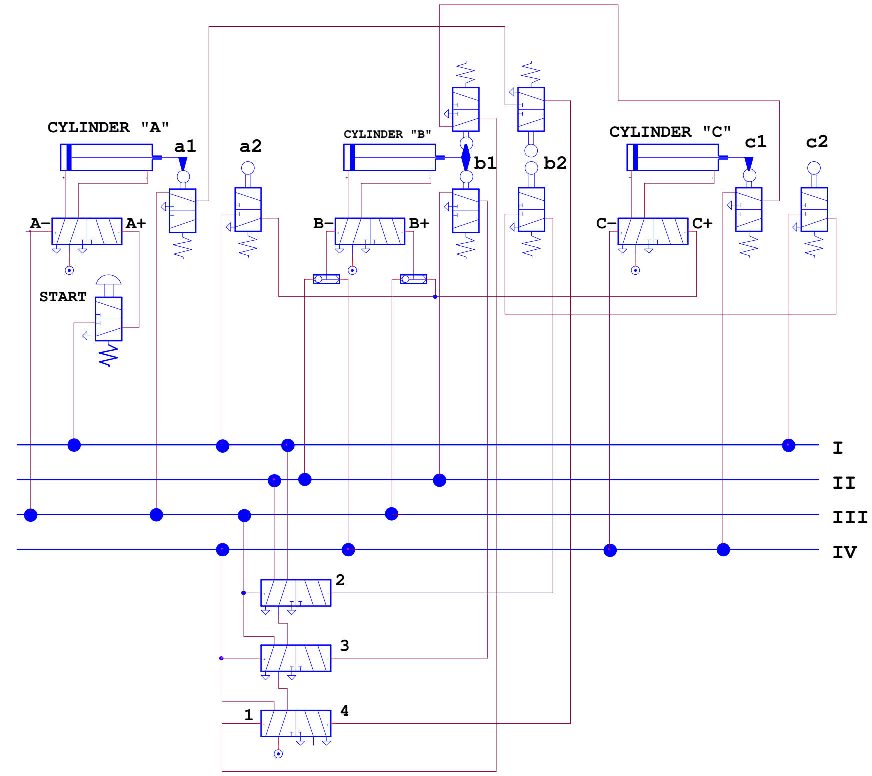
e. System Cylinders Requirements

Fig. 6-8 : Pneumatic Cascade Design Process Steps

6-4



f. System Components



g. Complete System

Fig. 6-8 : Pneumatic Cascade Design Process Steps

(Cont'd)

$$\text{START}, \begin{bmatrix}A+\\B+\end{bmatrix}, B-, A-, \begin{bmatrix}C+\\D+\end{bmatrix}, \begin{bmatrix}D-\\C-\end{bmatrix}$$

$$\text{START}, \underset{\text{I}}{\begin{bmatrix}A+\\B+\end{bmatrix}} \Bigg| \underset{\text{II}}{, B-, A-, \begin{bmatrix}C+\\D+\end{bmatrix},} \Bigg| \underset{\text{I}}{\begin{bmatrix}D-\\C-\end{bmatrix}}$$

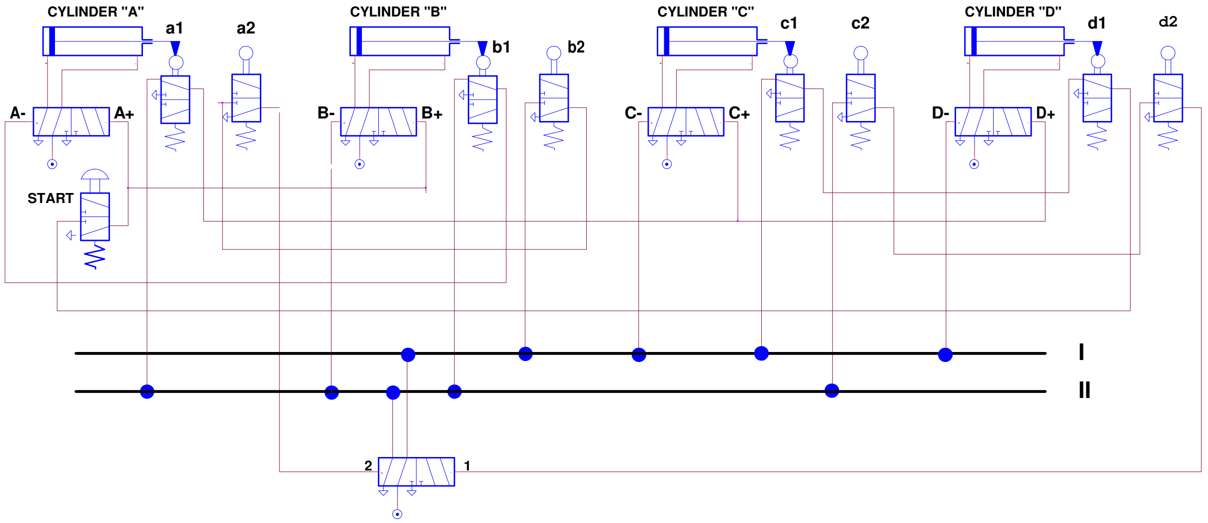


Fig. 6-9 : Pneumatic Cascade Example 1

(from semester exam September 2002)

$$\text{START}, A+, B+, \begin{bmatrix}A-\\C+\end{bmatrix}, \begin{bmatrix}B-\\C-\end{bmatrix}$$

$$\text{START}, \underset{\text{I}}{A+, B+,} \Bigg| \underset{\text{II}}{\begin{bmatrix}A-\\C+\end{bmatrix}} , \Bigg| \underset{\text{III}}{\begin{bmatrix}B-\\C-\end{bmatrix}} \Bigg|$$

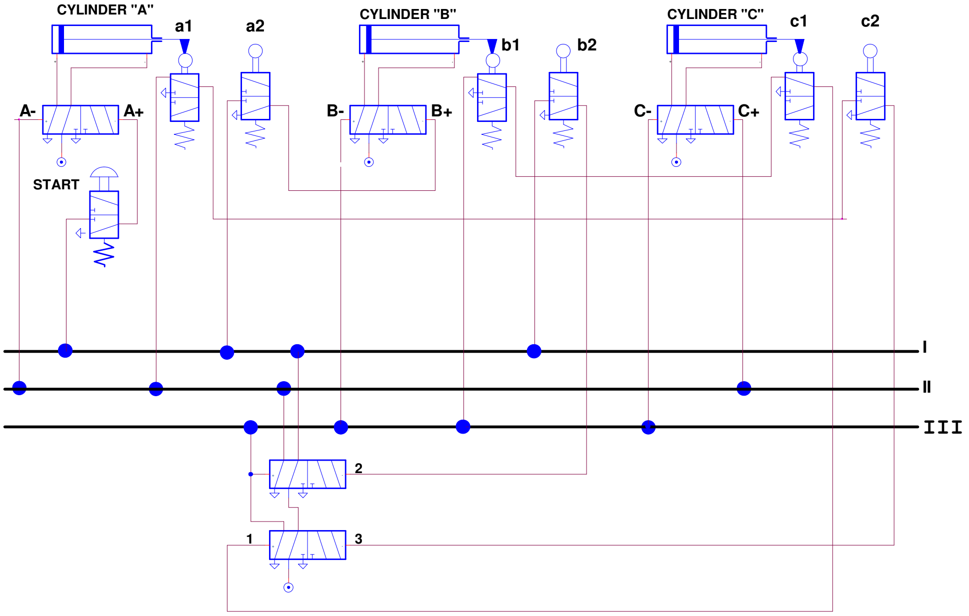


Fig. 6-10 : Pneumatic Cascade Example 2

(from semester exam July 2002)

Efficient Use of Directional-Control Valves in Fluid-Logic Circuits

Various types of directional-control valves have been systematically analized to determine the logic output functions obtained for different input-signal combinations. It was found that a surprisingly large variety of fairly complex logic functions can be obtained with a single valve.The results are summarised in five tables which include 186 (i.e. all the significant) input-signal combinations. Several examples are presented, illustrating how the tables can be used to implement given fluid-logic functions with a minimum number of valves.
It thus becomes possible to exploit the logic capacity of these valves to their fullest extent, which can result in considerable reduction in the number of valves.

D.W. Pessen
Associate Professor
Dept of Mechanical Engineering,
Technion - Israel Institute of Technology,
Haifa, Israel
Mem ASME

Fig. 6-11 : Function Tables Background

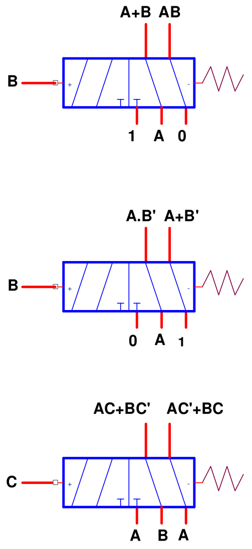


Fig. 7-19 : Examples of Complex Functions

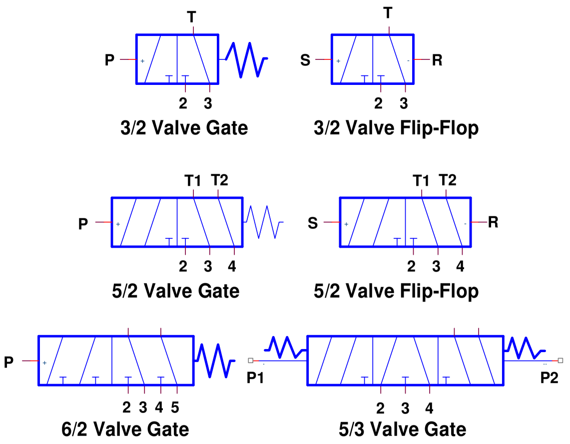
1	YES
2	NOT
3	AND
4	OR
5	INHIBITION
6	IMPLICATION
7	SELECTOR

a. Function Search Order

- a. 3/2 Valve , Single Output Function
- b. Two Separate Output Function, 2 Input Variables
- c. Two Separate Output Function, 3 Input Variables
- d. Two Separate Output Function, 4/5 Input Variables
- e. Two Separate Flip-Flop Output Function

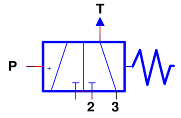
b. Tables List

Fig. 6-12 : Function Tables Arangement



a. Valves Pins Assignment

Note : Use of shuttle valves is uaually less expensive.



Line No.	Function Name	Boolean Function	P	2	3
1	YES	T = A	A	1	0
2	NOT	T = A'	A	0	1
3	AND	T = AB	B	A	0
4	AND	T = AB	B	A	B
5	OR	T = A+B	B	1	A
6	OR	T = A+B	B	B	A
7	INHIBITION	T = AB'	B	0	A
8	IMPLICATION	T = A+B'	B	A	1
9	SELECTOR	T = AC+BC'	C	A	B

b. 3/2 Valve Connections yielding Single Output Function

Fig. 6-13 : 3/2 Valve Connections yielding Single Output Function and Valves Pins Assignment

				5/2 Valve	6/2 Valve	5/3 Valve
No.	Function Names	Output 1	Output 2	P 2 3 4	P 2 3 4 5	P1 P2 2 3 4
10	YES/NOT	T2=A	T1=A’	A 0 1 0		
12	YES/AND	T2=B	T1=AB		B A 0 1 0	
13	YES/OR	T2=B	T1=A+B		B 1 A 1 0	
14	YES/INHIBITION	T1=B	T2=AB’	B 1 0 A		
16	YES/IMPLICATION	T2=B	T1=A+B’	B A 1 0		
18	NOT/AND	T2=B’	T1=AB	B A 0 1		
21	NOT/OR	T1=B’	T2=A+B	B 0 1 A		
24	NOT/INHIBITION	T1=B’	T2=AB’		B 0 1 0 A	
25	NOT/IMPLICATION	T1=B’	T2=A+B’		B 0 1 A 1	
26	AND/OR	T2=AB	T1=A+B	B 1 A 0		
27	AND/OR	T2=AB	T1=A+B	B B A 0		
36	AND/INHIBITION	T2=AB	T1=AB’	B 0 A 0		
42	AND/IMPLICATION	T1=AB	T2=A+B’			B A A 0 1
43	AND/IMPLICATION	T1=AB	T2=A+B’		B A 0 A 1	
45	OR/INHIBITION	T1=A+B	T2=AB’			A B A 1 0
46	OR/INHIBITION	T1=A+B	T2=AB’		B 1 A 0 A	
47	OR/INHIBITION	T1=A+B	T2=AB’		B B A 0 A	
48	OR/IMPLICATION	T2=A+B	T1=A+B’	B A 1 A		
54	INHIBITION/INHIBITION	T1=A’B	T2=AB’			A B 0 1 0
55	INHIBITION/IMPLICATION	T1=AB’	T2=A+B’	B 0 A 1		
58	IMPLICATION/IMPLICATION	T1=A+B’	T2=A’+B			A B 1 0 1

: Valve Connections Yielding Two Separate Outputs, Two Variables

Fig. 6-14

No.	FUNCTION NAME	BOOLEAN FUNCTION 1	BOOLEAN FUNCTION 2	Valve 5/2 P 2 3 4	Valve 6/2 P 2 3 4 5	Valve 5/3 P1 P2 2 3 4
59	YES/SELECTOR	T2=C	T1=AC+BC’		C A B 1 0	
60	NOT/SELECTOR	T2=C’	T1=AC+BC’		C A B 0 1	
61	AND/OR	T1=AB	T2=B+C	B A B C		
66	AND/AND	T1=AC	T2=BC		C A 0 B 0	
70	AND/INHIBITION	T1=AC	T2=BC’	C A 0 B		
73	AND/IMPLICATION	T1=AC	T2=B+C’		C A 0 B 1	
75	AND/SELECTOR	T2=BC	T1=AC+BC’	C A B 0		
79	AND/SELECTOR	T2=AC	T1=AC+BC’		C A B A 0	
81	OR/OR	T1=B+C	T2=A+C		C 1 B 1 A	
84	OR/OR	T1=B+C	T2=A+C		C C B C A	
85	OR/INHIBITION	T1=B+C	T2=AC’		C 1 B 0 A	
86	OR/INHIBITION	T1=B+C	T2=AC’		C C B 0 A	
87	OR/IMPLICATION	T2=A+C	T1=B+C’	C B 1 A		
90	OR/SELECTOR	T1=A+C	T2=AC+BC’	C 1 A B		
91	OR/SELECTOR	T1=A+C	T2=AC+BC’	C C A B		
94	OR/SELECTOR	T1=B+C	T2=AC+BC’		C 1 B A B	
95	OR/SELECTOR	T1=B+C	T2=AC+BC’		C C B A B	
96	INHIBITION/INHIBITION	T1=AC’	T2=BC’		C 0 A 0 B	
97	INHIBITION/INHIBITION	T1=AB’C	T2=ABC’			B C 0 A 0
99	INHIBITION/IIMPLICATION	T1=AC’	T2=B+C’		C 0 A B 1	
100	INHIBITION/IIMPLICATION	T1=AB’C	T2=A+B’+C			B C 0 A 1
101	INHIBITION/SELECTOR	T1=AC’	T2=AC+BC’	C 0 A B		
103	INHIBITION/SELECTOR	T1=BC’	T2=AC+BC’		C 0 B A B	
104	IMPLICATION/IMPLICATION	T1=A+C’	T2=B+C’		C A 1 B 1	
105	IMPLICATION/IMPLICATION	T1=A+B+C’	T2=A+B’+C			B C 1 A 1
107	IMPLICATION/SELECTOR	T2=B+C’	T1=AC+BC’	C A B 1		
109	IMPLICATION/SELECTOR	T2=A+C’	T1=AC+BC’		C A B A 1	
110	SELECTOR/SELECTOR	T2=AC+BC’	T1=AC’+BC	C A B A		
112	AND+AND/AND+AND	T1=AB+AC	T2=AC+BC			C A A B C
113	AND+AND/AND+	T1=AB+AC	T2=A+BC			C A A B A
114	AND+AND/INHIBITION	T1=AB+AC	T2=A’BC			C A A B 0
115	AND+AND/IMPLICATION	T1=AB+AC	T2=A+B+C’			C A A B 1
117	AND+INHIBITION/AND+INHIBITION	T1=AB+AC’	T2=AB’+AC			B C A 0 A
118	AND+INHIBITION/INHIBITION+	T1=AB+AC’	T2=A+BC’			B C A B A
119	AND+INHIBITION/OR	T1=AB+AC’	T2=B+C			B C A B C
120	AND+INHIBITION/AND	T1=AB+AC’	T2=BC			B C A 0 B
121	AND+INHIBITION/INHIBITION	T1=AB+AC’	T2=BC’			B C A B 0
122	AND+INHIBITION/IMPLICATION	T1=AB+AC’	T2=B’+C			B C A 0 1
127	AND+/AND+	T1=A+BC	T2=C+AB			A C A B C
128	AND+/INHIBITION	T1=A+BC	T2=ABC’			A C A B 0
129	AND+/IMPLICATION	T1=A+BC	T2=A’+B+C			A C A B 1
131	INHIBITION+/INHIBITION+	T1=A+BC’	T2=A+B’C			C B A 1 A
132	INHIBITION+/OR	T1=A+BC’	T2=B+C			C B A 1 B
133	INHIBITION+/AND	T1=A+BC’	T2=BC			C B A B C
134	INHIBITION+/INHIBITIOM	T1=A+BC’	T2=B’C			C B A 1 0
135	INHIBITION+/IMPLICATION	T1=A+BC’	T2=B+C’			C B A B 1

Fig 6-15 : Valve Connections Yielding 2 Separate Output Functions, 3 Variables

				5/2 Valve	6/2 Valve	5/3 Valve
No.	Function Names	Output 1	Output 2	P 2 3 4	P 2 3 4 5	P1 P2 2 3 4
	5 VARIABLES					
140		T1=AE+BE’	T2=CE+DE’		E A B C D	
141		T1=AB+AC’+B’CD	T2=CE+B’E+BC’D			B C A D E
	4 VARIABLES					
142	AND/SELECTOR	T2=CD	T1=AD+BD’		D A B C 0	
143	AND/SELECTOR	T2=CD	T1=AD+BD’		D A B C D	
144	OR/SELECTOR	T2=C+D	T1=AD+BD’		D A B 1 C	
145	OR/SELECTOR	T2=C+D	T1=AD+BD’		D A B D C	
146	INHIBITION/SELECTOR	T2=CD’	T1=AD+BD’		D A B 0 C	
147	IMPLICATION/SELECTOR	T2=C+D’	T1=AD+BD’		D A B C 1	
148	SELECTOR/SELECTOR	T1=AD+BD’	T2=BD+CD’	D A B C		
149	SELECTOR/SELECTOR	T1=AD+BD’	T2=CD+BD’		D A B C B	
150	SELECTOR/SELECTOR	T1=AD+BD’	T2=CD+AD’		D A B C A	
151	SELECTOR/SELECTOR	T1=AD+BD’	T2=AD+CD’		D A B A C	
152	AND+INHIBITION/AND+INHIBITION	T1=AB+AC’	T2=CD+B’D			B C A 0 D
153	AND+INHIBITION/INHIBITION+	T1=A+B’C	T2=CD+B’D			B C A C D
154	INHIBITION+/INHIBITION+	T1=A+B’C	T2=D+BC’			B C A 1 0
155		T1= AB+AC’+B’CD	T2=C+BD			B C A D C
156		T1= AB+AC’+B’CD	T2=BC+BD			B C A D B
157		T1= AB+AC’+B’CD	T2=BC’D			B C A D 0
158		T1=AB+AC’+B’CD	T2=B’+C+D			B C A D 1
159		T1=AB+AC’+B’CD	T2=AB’+AC+B’CD			B C A D A
160		T1=AB+AC’+B’CD	T2=D			B C A D D

[redacted] **: Valve Connections Yielding Two Separate Outputs, 4-5 Variables**

Fig. 6-16

			3/2 Valve	5/2 Valve
No.	Output 1	Output 2	2 3	2 3 4
	No Input Variables			
161	T=y		1 0	
162	T1=y’	T2=y		0 1 0
163	T1=y	T2=y’		1 0 1
	One Input Variables			
164	T=Ay		A 0	
165	T=Ay’		0 A	
166	T1=Ay’	T2=Ay		0 A 0
167	T1=Ay	T2=Ay’		A 0 A
168	T1=Ay	T2=y’		A 0 1
169	T1=y	T2=Ay’		1 0 A
170	T=A+y’		A 1	
171	T=A+y		1 A	
172	T1=A+y	T2=A+y’		1 A 1
173	T1=A+y’	T2=A+y		A 1 A
174	T1=A+y’	T2=y		A 1 0
175	T1=y’	T2=A+y		0 1 A
176	T1=Ay’	T2=A+y’		0 A 1
177	T1=A+y	T2=Ay		1 A 0
	Two Input Variables			
178	T1=Ay	T2=By’		A 0 B
179	T1=A+y’	T2=B+y		A 1 B
180	T1=Ay+By’		A B	
181	T1=Ay+By’	T2=By		A B 0
182	T1=Ay+By’	T2=B+y’		A B 1
183	T1=Ay’	T2=Ay+By’		0 A B
184	T1=A+y	T2=Ay+By’		1 A B
185	T1=Ay+By’	T2=By+Ay’		A B A
	Three Input Variables			
186	T1=Ay+By’	T2=By+Cy’		A B C

Fig. 6-17 : Valve Connections Yielding Flip-Flop Output Functions

Required Functions

F1 = X1.X2'
F2 = X1.X2 + X2'.X3

F1 is Inhibition Function
F2 is Selector Function
Search in 3-variables table
Found Functions 101 and 103
Only 101 satisfies requirements

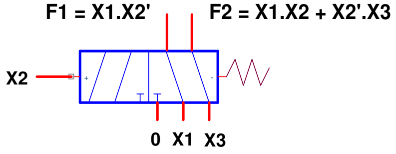
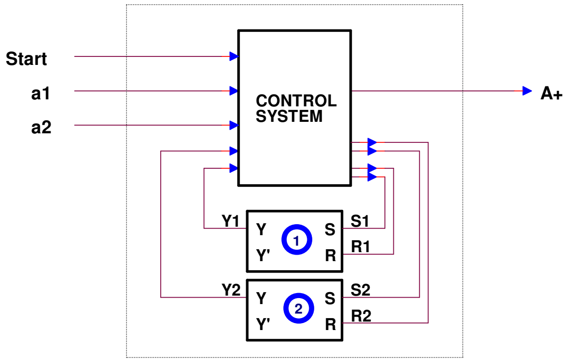


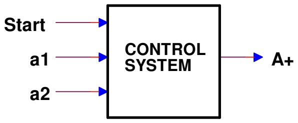
Fig. 6-18 : Example 1



d. System Solution Block Diagram
(Huffman Method Realization)

START,A+,A-,A+,A-

a. Process Sequence



b. System Block Diagram

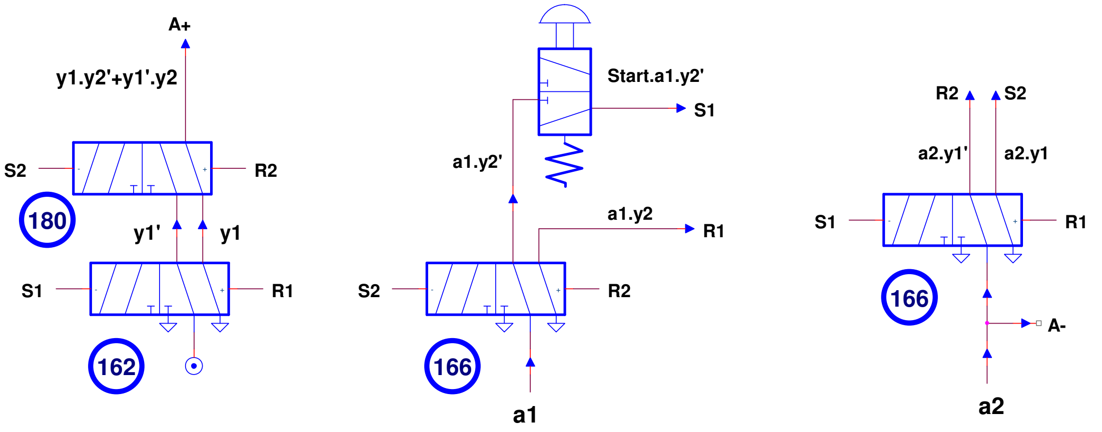
S1=Start.a1.y2'   R1=a1.y2
S2=a2.y1   r2=a2.y1'

A+ = y1.y2'+y1'.y2
A- = a2

C. Huffman Method Solution

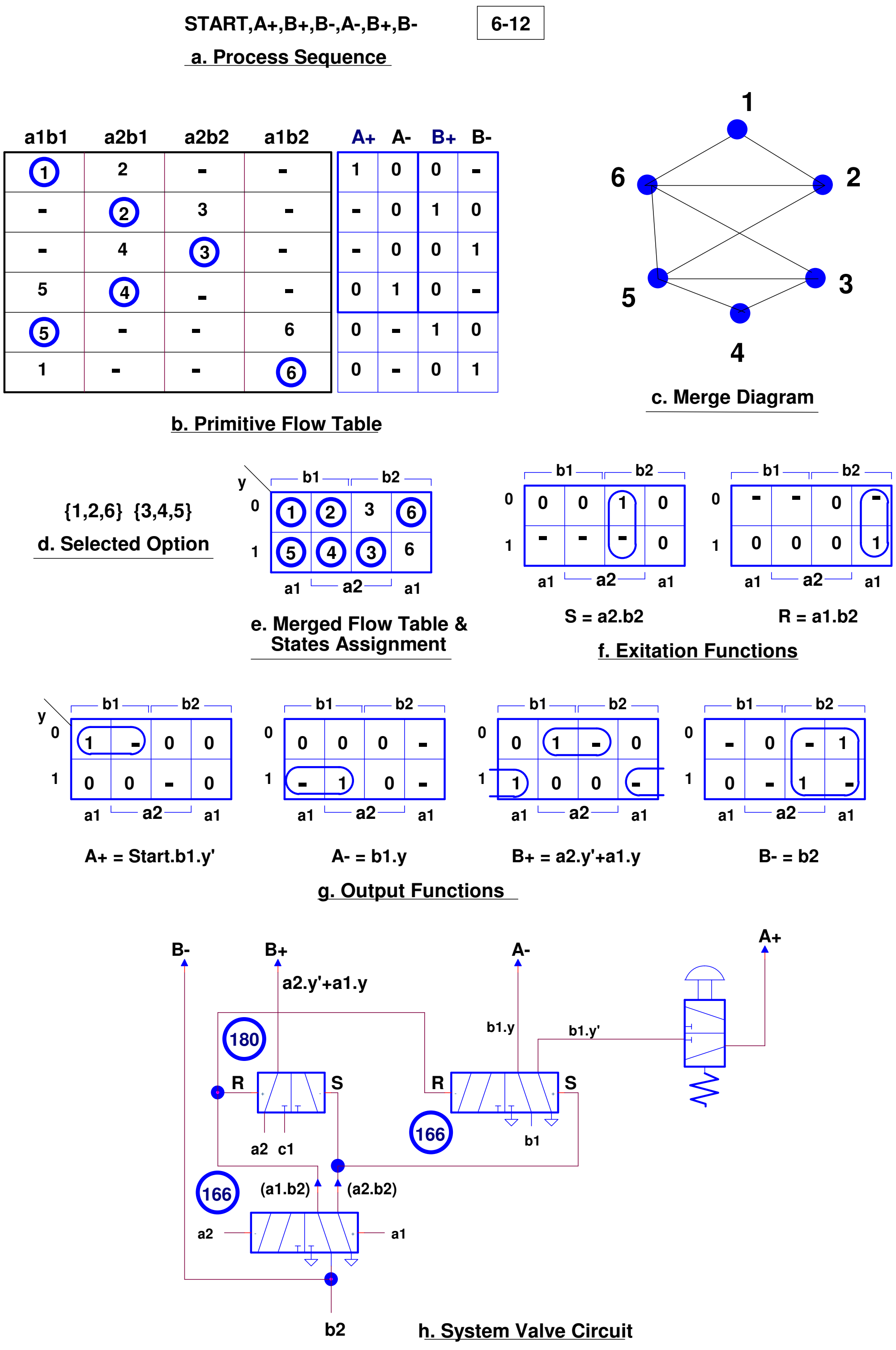
- * Start , a1, a2 are external signals
- * y1 ,y2 are outputs of internal flip-flops
- * S1,R1,S2,R2 are Set/Reset of those flip-flops
- * Functions that contain only external variables are to be searched in first 4 tables
- * Function that contain internal variables (y) are to be searchedfirst in fifth table
- * In current system, all functions contain internal variables, so they are to be searched in fifth table

e. Design Considerations



f. System Valve Circuit

Fig. 6-19 : Example 2



START,A+,B+,B-,A-,B+,B-

6-12

a. Process Sequence

a1b1	a2b1	a2b2	a1b2	A+	A-	B+	B-
①	2	-	-	1	0	0	-
-	②	3	-	-	0	1	0
-	4	③	-	-	0	0	1
5	④	-	-	0	1	0	-
⑤	-	-	6	0	-	1	0
1	-	-	⑥	0	-	0	1

b. Primitive Flow Table

c. Merge Diagram

{1,2,6} {3,4,5}

d. Selected Option

y	a1b1	a2b1	a2b2	a1b2
0	①	②	3	⑥
1	⑤	④	③	6

e. Merged Flow Table & States Assignment

y	a1b1	a2b1	a2b2	a1b2
0	0	0	1	0
1	-	-	-	0

S = a2.b2

y	a1b1	a2b1	a2b2	a1b2
0	-	-	0	-
1	0	0	0	1

R = a1.b2

f. Exitation Functions

y	a1b1	a2b1	a2b2	a1b2
0	1	-	0	0
1	0	0	-	0

A+ = Start.b1.y'

y	a1b1	a2b1	a2b2	a1b2
0	0	0	0	-
1	-	1	0	-

A- = b1.y

y	a1b1	a2b1	a2b2	a1b2
0	0	1	-	0
1	1	0	0	-

B+ = a2.y'+a1.y

y	a1b1	a2b1	a2b2	a1b2
0	-	0	-	1
1	0	-	1	-

B- = b2

g. Output Functions

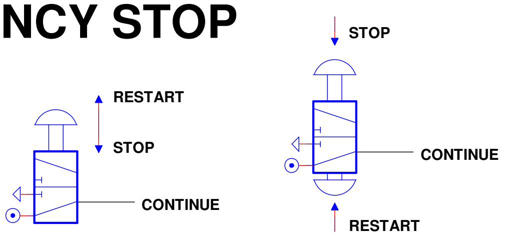
h. System Valve Circuit

Fig. 6-20 : Solution for Sequence START,A+,B+,B-,A-,B+,B-

EMERGENCY STOP

Stop process due to emergency condition
Easy accessible buttons
Safety continuation Restart (Continue)
Different modes
Add-on

a. Target & Definition



b. Single STOP-RESTART button

c. Separate STOP/RESTART buttons

Fig. 6-21 : STOP-RESTART Single/Separate Buttons

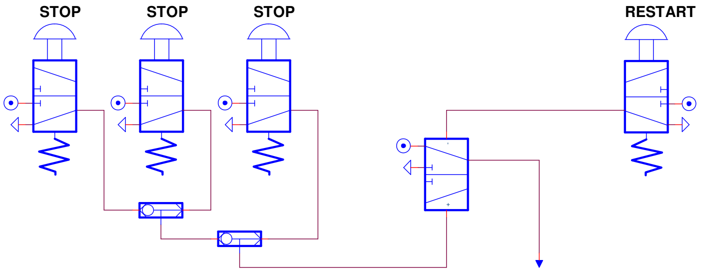


Fig. 6-22 : STOP-RESTART With Multiple Remote-Control STOP Buttons

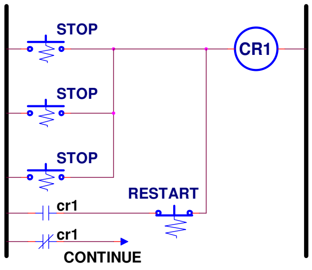


Fig. 6-23 : Electric STOP-RESTART System With Multiple Remote-Control STOP Buttons

No Change

Cylinder at rest must remain at rest.
Cylinder in motion must complete its stroke, and remain at rest.

No Motion

Cylinder at rest must remain at rest.
Cylinder in motion must be disconnected from pressure and move until stopped by friction (or end stroke).

Lock Piston

Cylinder at rest must remain at rest.
Cylinder in motion must be locked at its current position.

Safety Position

Cylinder must be moved to its either (+) or (-) position, which been pre-definrd as "Safety Position".

No-Change/No-Motion

Combination of No-Change and No-Motion modes.

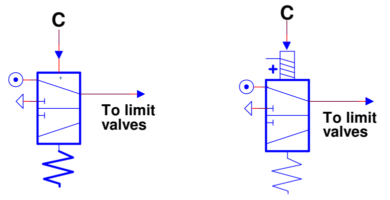
No-Change/Lock-Piston

Combination of No-Change and Lock-Piston modes.

No-Change/Safety-Position

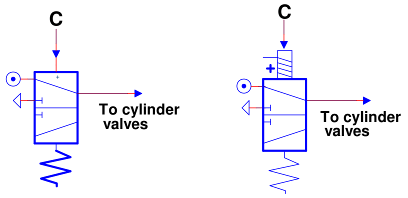
Combination of No-Change and Safety-Position modes.

Fig. 6-24 : Emergency Stop Modes Definition



a. No-Change Mode Pneumatic Actuation

b. No-Change Mode Electric Actuation



c. No-Motion Mode Pneumatic Actuation

d. No-Motion Mode Electric Actuation

Fig. 6-25 : No-Change & No-Motion Circuits

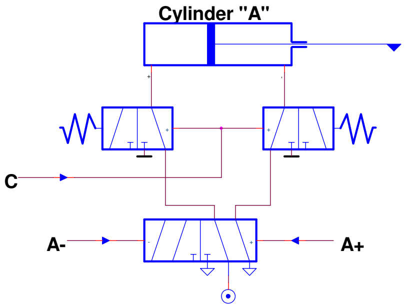


Fig. 6-26 : "Lock-Piston" Mode Circuit

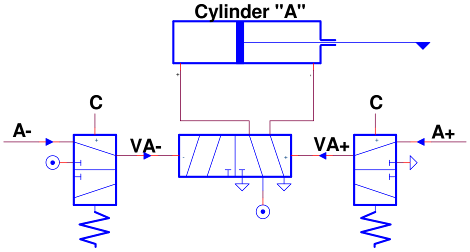


Fig. 6-27 : "Safety-Position" Mode Circuit
(Applied to an Individual Cylinder)

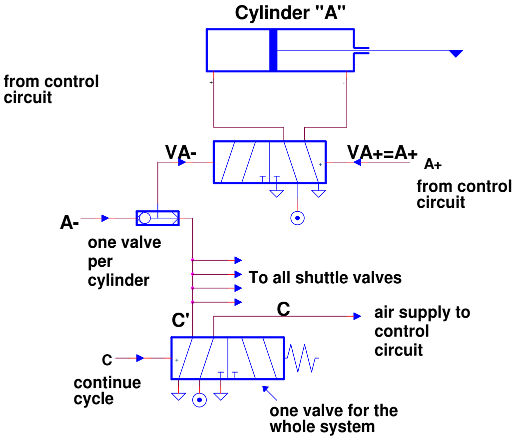


Fig. 6-28 : "Safety-Position" Mode Circuit
(Applied to Entire System)

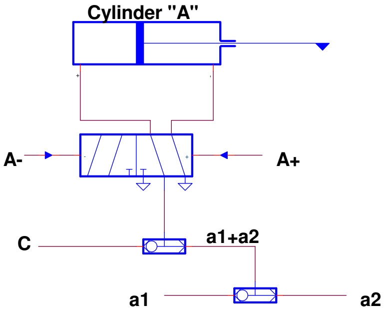
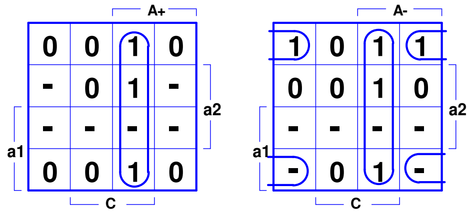


Fig. 6-29 : "No-Change/No-Motion" Mode Circuit



0	0	1	0
-	0	1	-
-	-	-	-
0	0	1	0

1	0	1	1
0	0	1	0
-	-	-	-
-	0	1	-

VA+ = C(A+)

VA- = C(A-)+C'.a2'

Fig. 6-30 : Karnaugh Maps for "No-Change/Safety-Position"Circuit

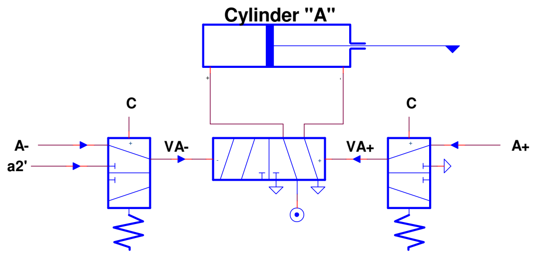


Fig. 6-31 : "No-Change/Safety-Position" Mode Circuit

CHAPTER 7

HARDWARE PROGRAMMERS

Drum Programmer
Programmable Counter 1/n
Programmable Counter 2/n

7-1

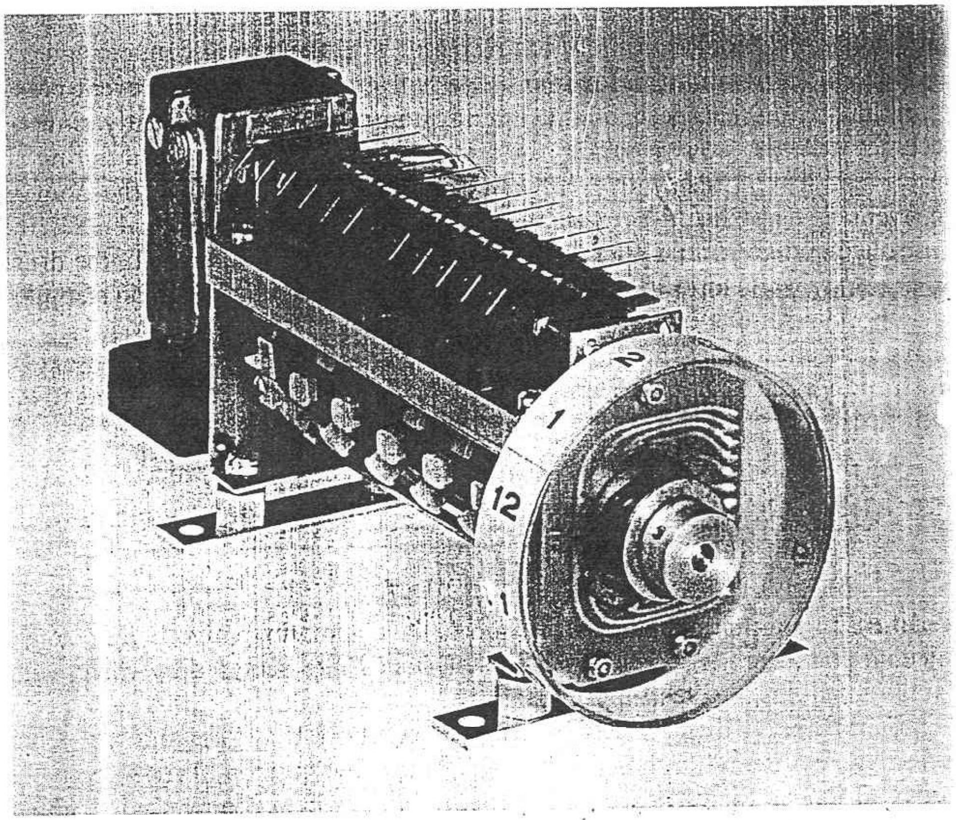
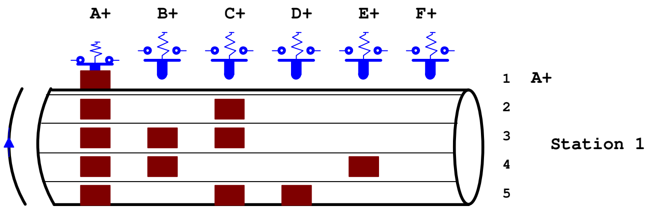


Fig. 7-1 : Drum programmer. (Courtesy of Amerace Corp.)

7-2

Step	1	2	3	4	5	6	7	....
Action	A+	C+	B+	C− E+	C+ D+ B−	F+	A− E−	....

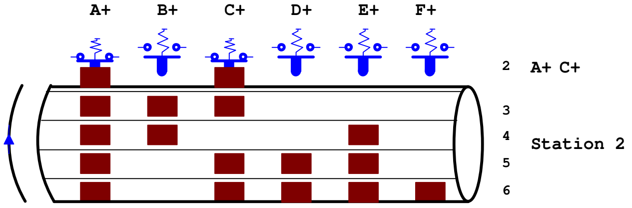
b. Sequence Flow



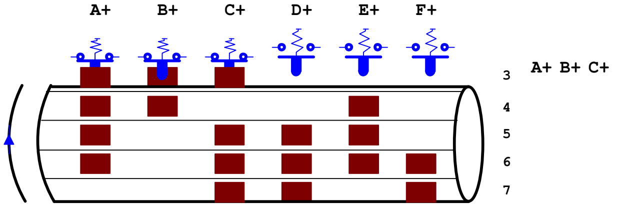
d. Station 1 Position

Station	A+	B+	C+	D+	E+	F+
1	■					
2	■		■			
3	■	■	■			
4	■	■			■	
5	■		■	■	■	
6	■		■	■	■	■
7			■	■		■
. . .						

c. Drum arrangement



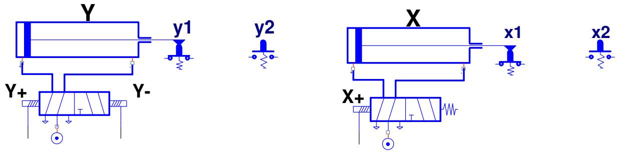
e. Station 2 Position



f. Station 3 Position

Fig. 7-2 : Drum Programmer Action

DRUM PROGRAMMER (STEPPING SWITCHED)



Cylinders D,E,F Types

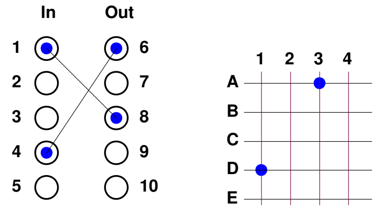
Cylinders A,B,C Types

$$\text{START}, A+, B+, B-, C+, C-, D+, A-, \begin{bmatrix} E+ \\ F+ \end{bmatrix}, F-, F+, F-, F+, \begin{bmatrix} D- \\ E- \\ F- \end{bmatrix}$$

Step	Desired Event	Signal to Advance	Switches									
			I	II	III	IV	V	VI	VII	VIII	IX	X
			Connected to									
			A+	B+	C+	D+	D-	E+	E-	F+	F-	Motor
1	/	Start										
2	A+	a2	X									
3	B+	b2	X	X								
4	B-	b1	X									
5	C+	c2	X		X							
6	C-	c1	X									
7	D+	d2	X			X						
8	A-	a1										
9	E+ F+	e2 f2						X		X		
10	F-	f1									X	
11	F+	f2								X		
12	F-	f1									X	
13	F+	f2								X		
14	D- E- F-	d1 e1 f1					X		X		X	
15-24	Go Home	/										X

Fig. 7-3 : Drum Programmer Solution

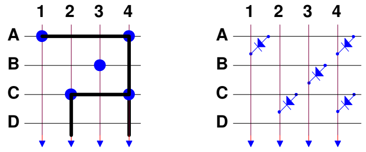
MATRIX BOARDS



a. Patch Board

b. Matrix Board

Fig. 7-4 : Boards



a. With Sneak Path

b. Without Sneak Path

Fig. 7-5 : Matrix Board

PROGRAMMABLE COUNTERS (SEQUENCERS)

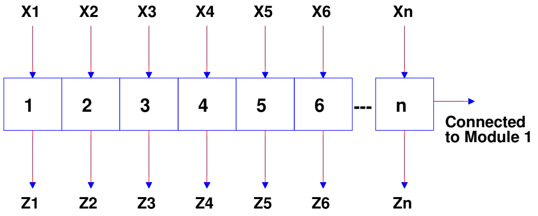


Fig. 7-6 : Schematic Representation of Programmable Counter

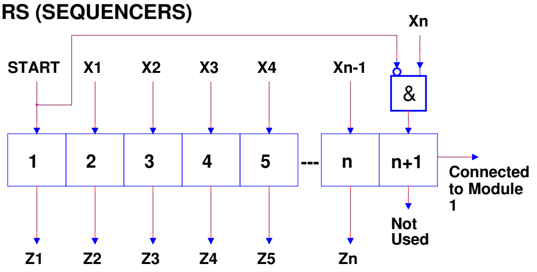
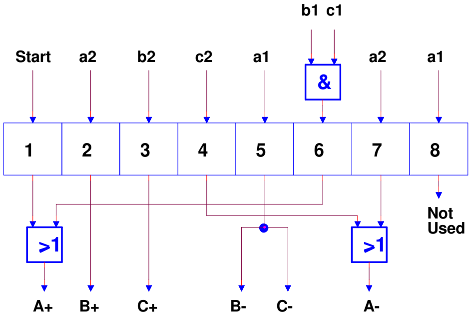


Fig. 7-7 : Programmable Counter With Single-Cycle START mode



$$\text{START}, A+, B+, C+, A-, \begin{bmatrix} B- \\ C- \end{bmatrix}, A+, A-$$

Fig. 7-8 : Programmable Counter Circuit for a Sequence, Implenting Actuating Valves Without Return Springs (1/n)

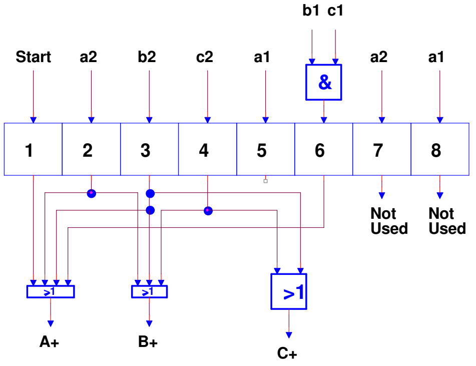
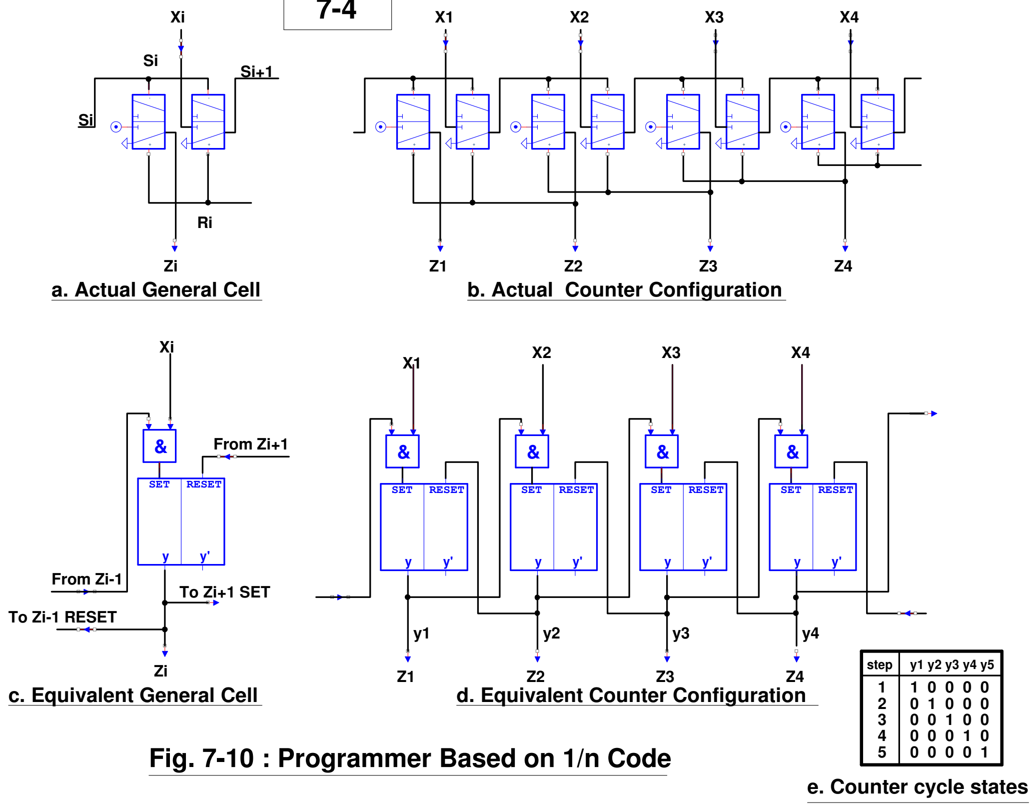


Fig. 7-9 : Programmable Counter Circuit for Sequence of Fig 7-8, While Actuating Valves Have Return Springs (1/n)



a. Actual General Cell

b. Actual Counter Configuration

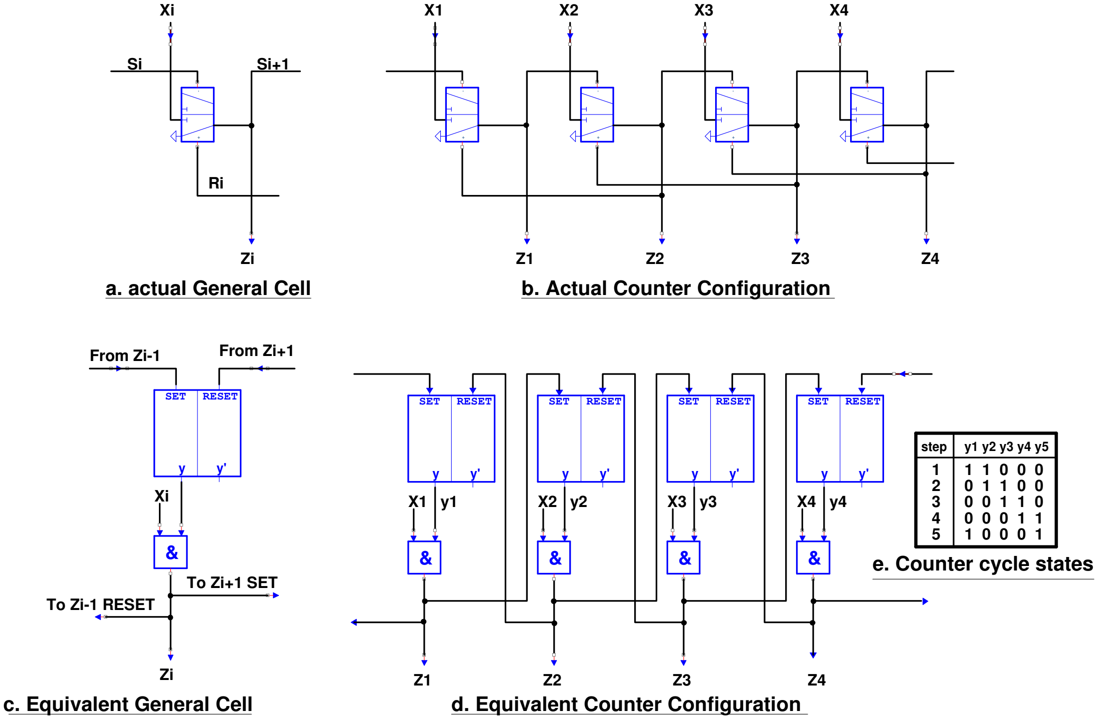
c. Equivalent General Cell

d. Equivalent Counter Configuration

step	y1 y2 y3 y4 y5
1	1 0 0 0 0
2	0 1 0 0 0
3	0 0 1 0 0
4	0 0 0 1 0
5	0 0 0 0 1

e. Counter cycle states

Fig. 7-10 : Programmer Based on 1/n Code



a. actual General Cell

b. Actual Counter Configuration

c. Equivalent General Cell

d. Equivalent Counter Configuration

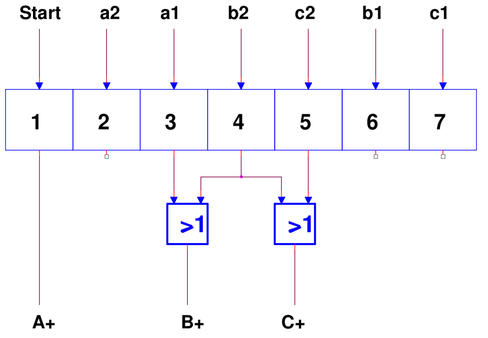
step	y1 y2 y3 y4 y5
1	1 1 0 0 0
2	0 1 1 0 0
3	0 0 1 1 0
4	0 0 0 1 1
5	1 0 0 0 1

e. Counter cycle states

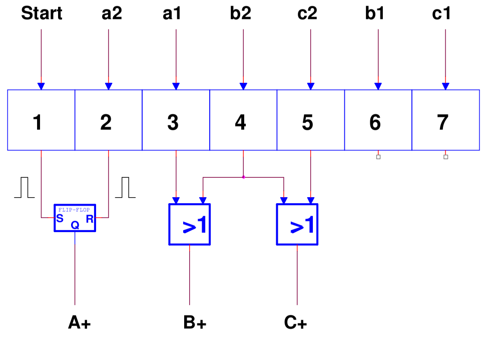
Fig. 7-11 : Programmer Based on 2/n Code

START , A+ , A- , B+ , C+ , B- . C-

a. Sequence



b. 1/n Prog.Counter Circuit



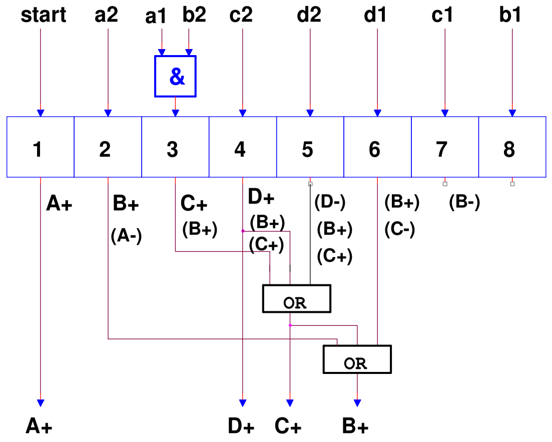
c. 2/n Programmable Counter Circuit

Fig. 7-12 : Programmable Counter Example 1

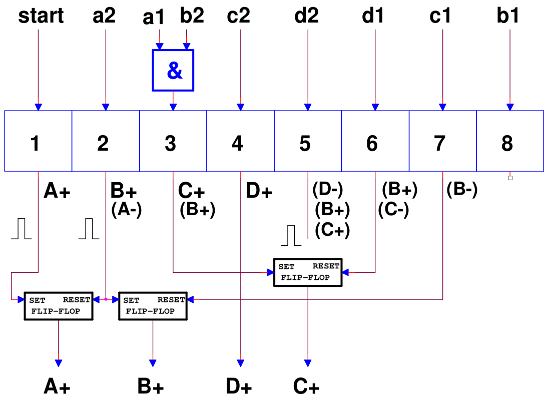
(With Return Springs)

START A+ , $\begin{bmatrix} B+ \\ A- \end{bmatrix}$, C+ , D+ , D- , C- , B-

a. Sequence



b. 1/n Prog.rammable Counter Circuit



c. 2/n Programmable Counter Circuit

Fig. 7-13 : Programmable Counter Example 2

(With Return Springs)

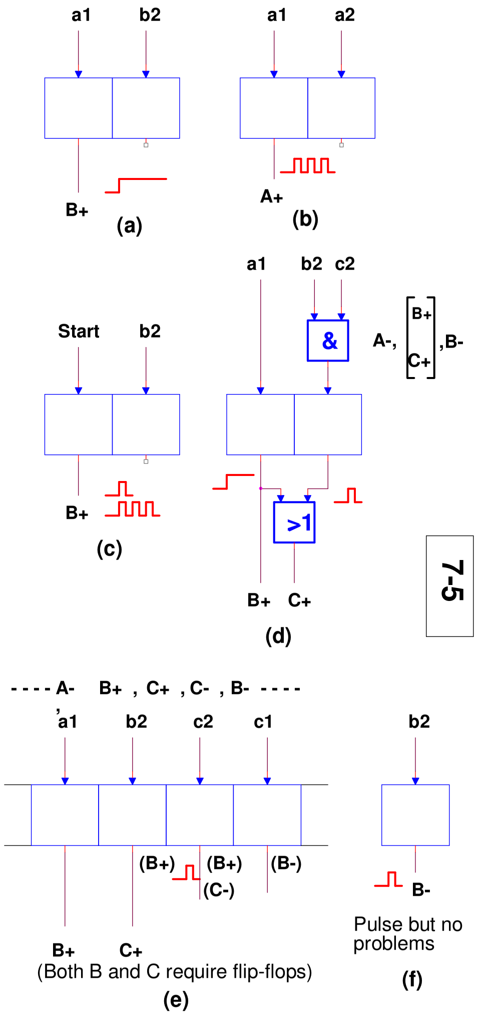


Fig. 7-14 : 2/n - Different Stability Cases

(With Return Springs)

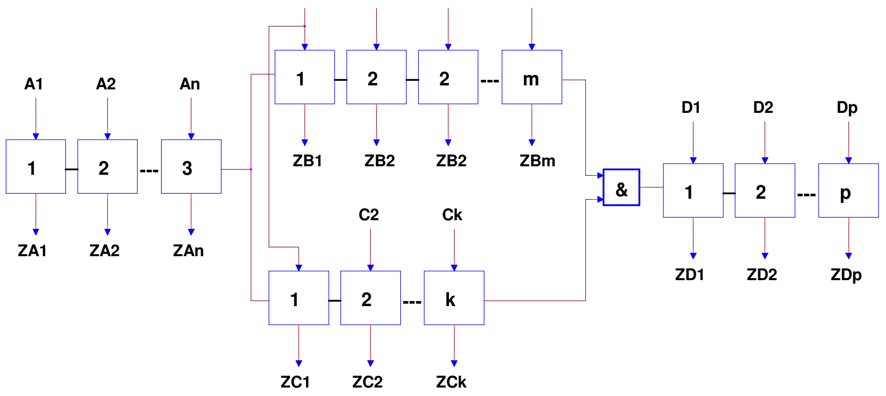
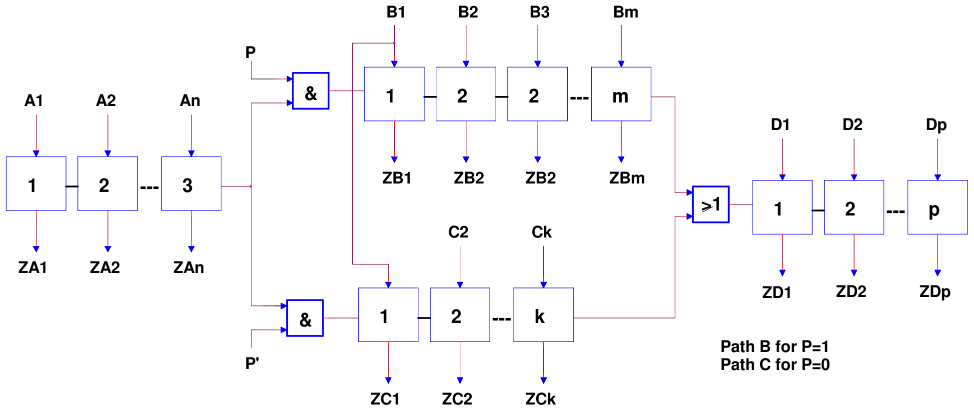
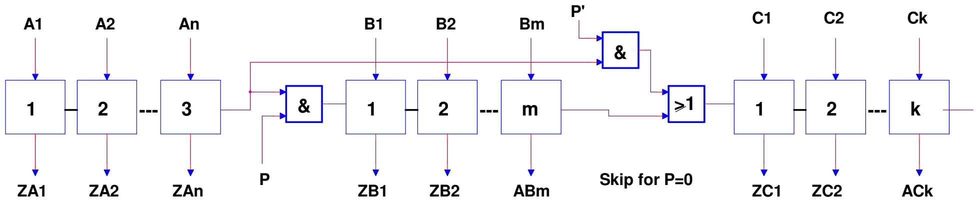


Fig. 7-15 : 1/n Programmer fot Two Simultaneous Parallel Paths



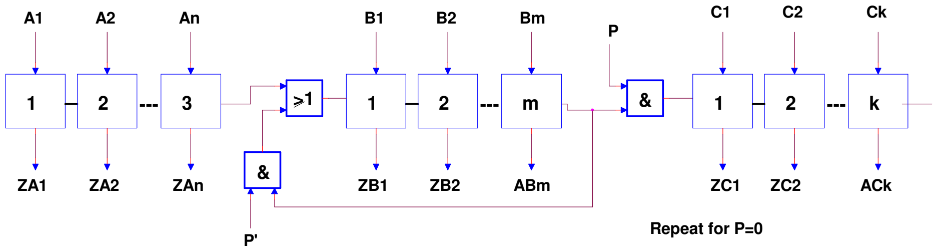
Path B for P=1
Path C for P=0

Fig. 7-16 : 1/n Programmer for Two Alternative Parallel Paths (Parallel)



Skip for P=0

Fig. 7-17 : 1/n Programmer for Program With Skip Steps Option



Repeat for P=0

Fig. 7-18 : 1/n Programmer for Program With Repeated Steps Option

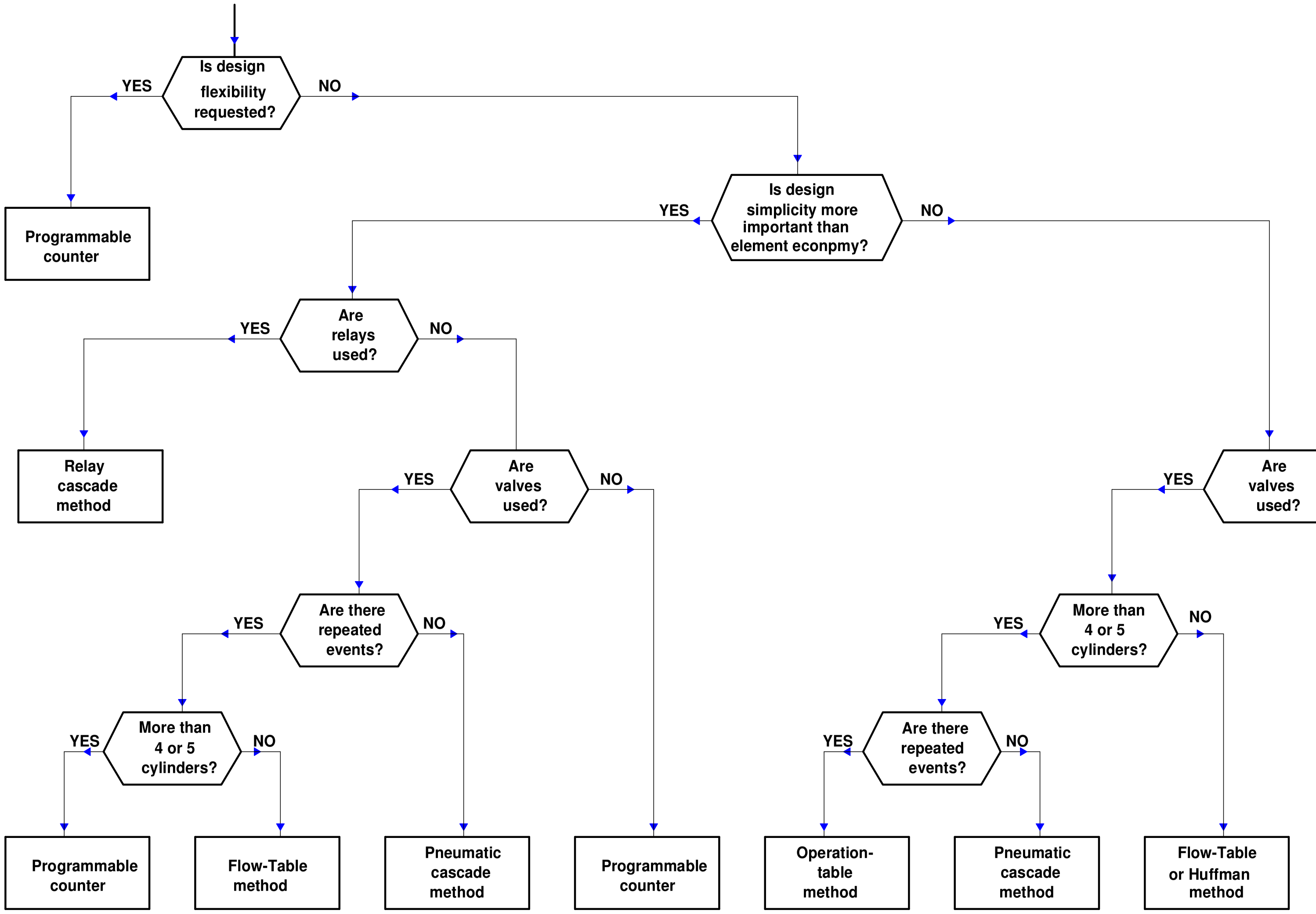


Fig. 7-19 : Flow Chart for Selecting Sequence-Control System Design Method

CHAPTER 8

MISCELLANEOUS SWITCHING ELEMENTS AND SYSTEMS

Timers Modes
Timers Construction
Pulse Shapers
Flip-Flop Types
Up/Down Counters
Shift Registers Implementation
Schmitt Trigger
Binary Codes
Error Detection
Encoders

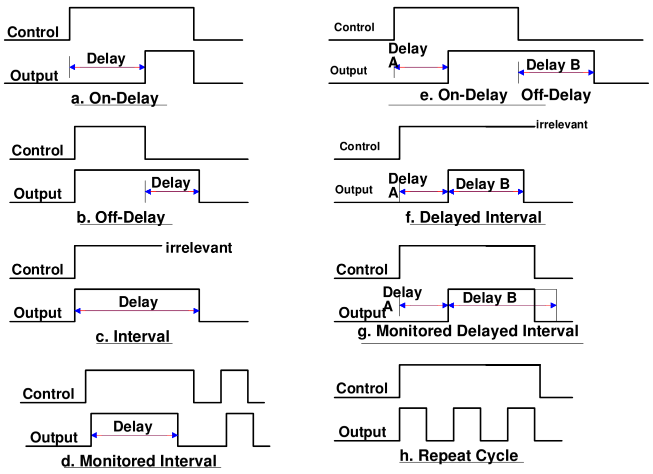


Fig. 8-1 : Sequence Chart of Common Timer Modes

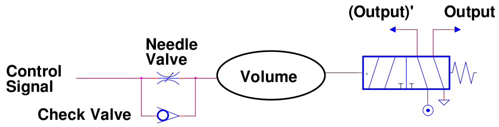


Fig. 8-2 : Pneumatic "On-Delay" Timer

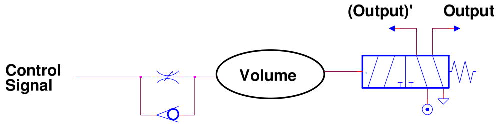


Fig. 8-3 : Pneumatic "Off-Delay" Timer

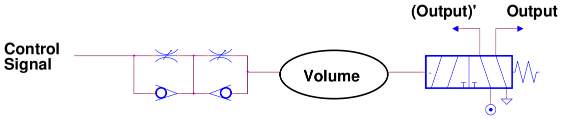


Fig. 8-4 : Pneumatic "On-Delay Off-Delay" Timer

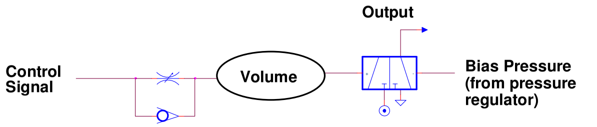


Fig. 8-5 : Pneumatic "On-Delay" Timer Using Bias Pressure

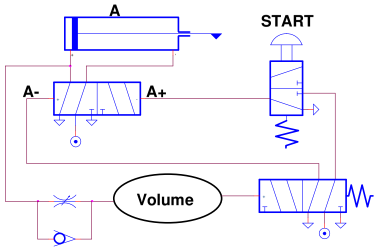


Fig. 8-6 : Implement "On-Delay" Timer for Automatic Cylinder Retraction (Interval Delay)

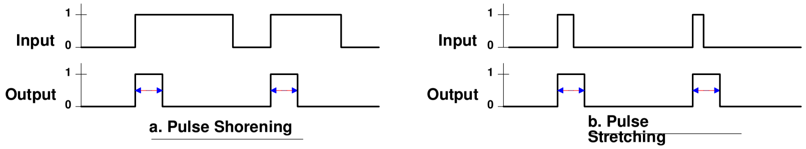


Fig. 8-7 : Sequence Chart for a Pulse Shaper (Monostable, One-Shot)

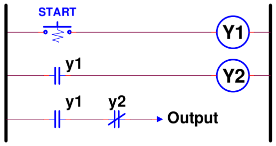


Fig. 8-8 : Relay Pulse Shaper

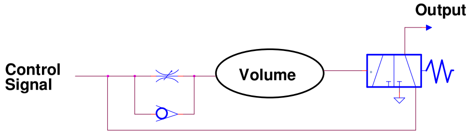
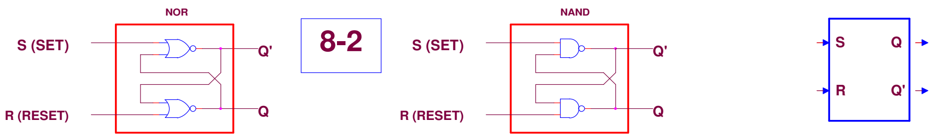


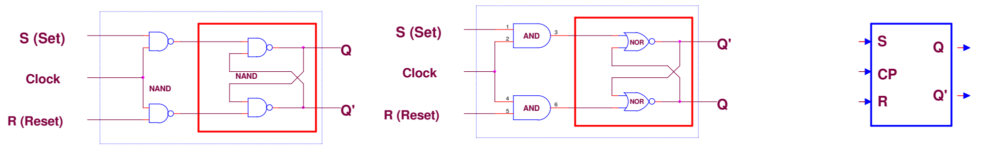
Fig. 8-9 : Pneumatic Pulse Shaper



a. Logic Circuit (Active HIGH)

b. Logic Circuit (Active LOW)

c. Short Symbol (Asynchronous FF)

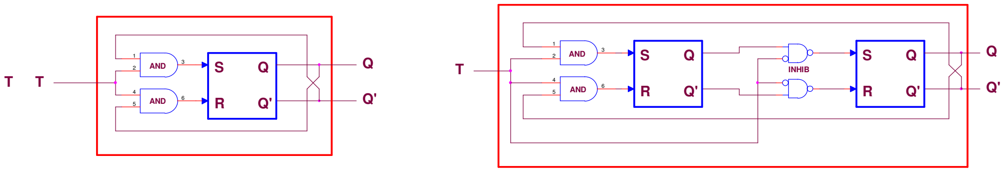


d. Clocked SR Flip-Flop (Synchronous)

e. Clocked SR Flip-Flop (Synchronous)

f. Short Symbol of Clocked Flip-flop

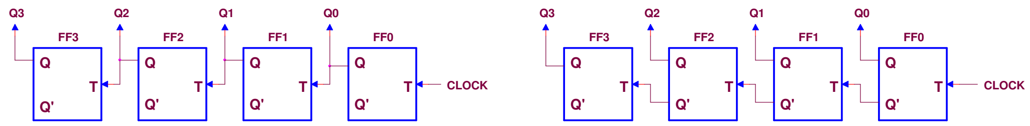
Fig. 8-10 : Basic Set-Reset Flip-Flop (SR)



a. Basic Logic Circuit

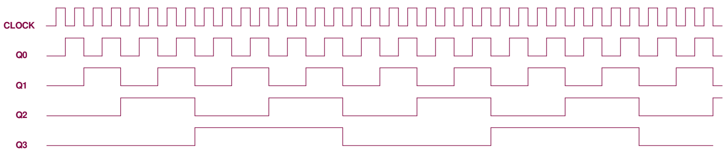
b. Master-Slave Type

Fig. 8-11 : Toggle Flip-Flop (T)



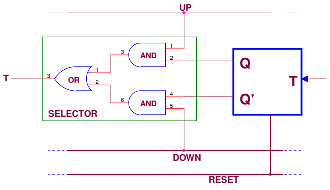
a. "UP" Counter

c. "DOWN" Counter



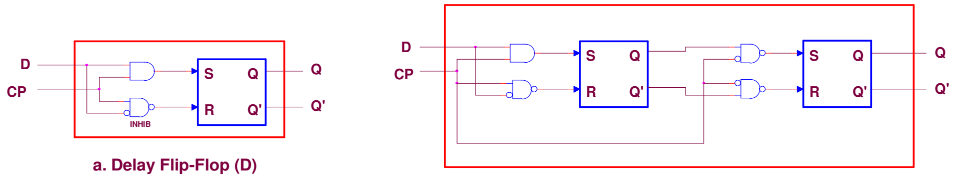
Q3 : 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 . . .
Q2 : 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 . . .
Q1 : 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 . . .
Q0 : 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 . . .

b. "UP" Counter Timing Waveforms



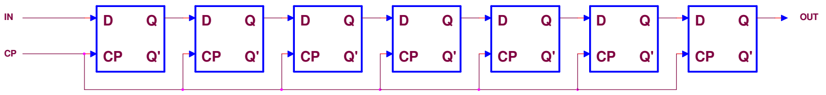
d. "UP/DOWN" Selector

Fig. 8-12 : T Flip-Flop Implementation - Counters
(Flip-Flops are Triggered on High-to-Low Transition at T)



a. Delay Flip-Flop (D)

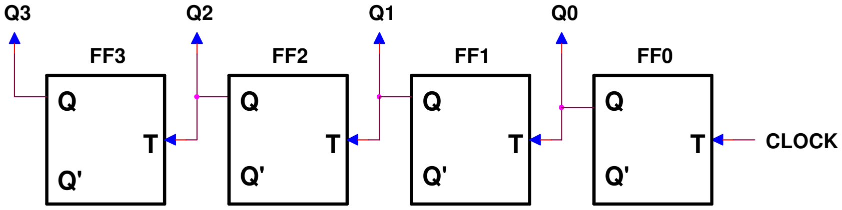
b. Master-Slave Type



c. Shift Registerr Based on D Flip-Flops

Fig. 8-13 : D Flip-Flop Implementation - Shift Register

8-3



a. "UP" Counter

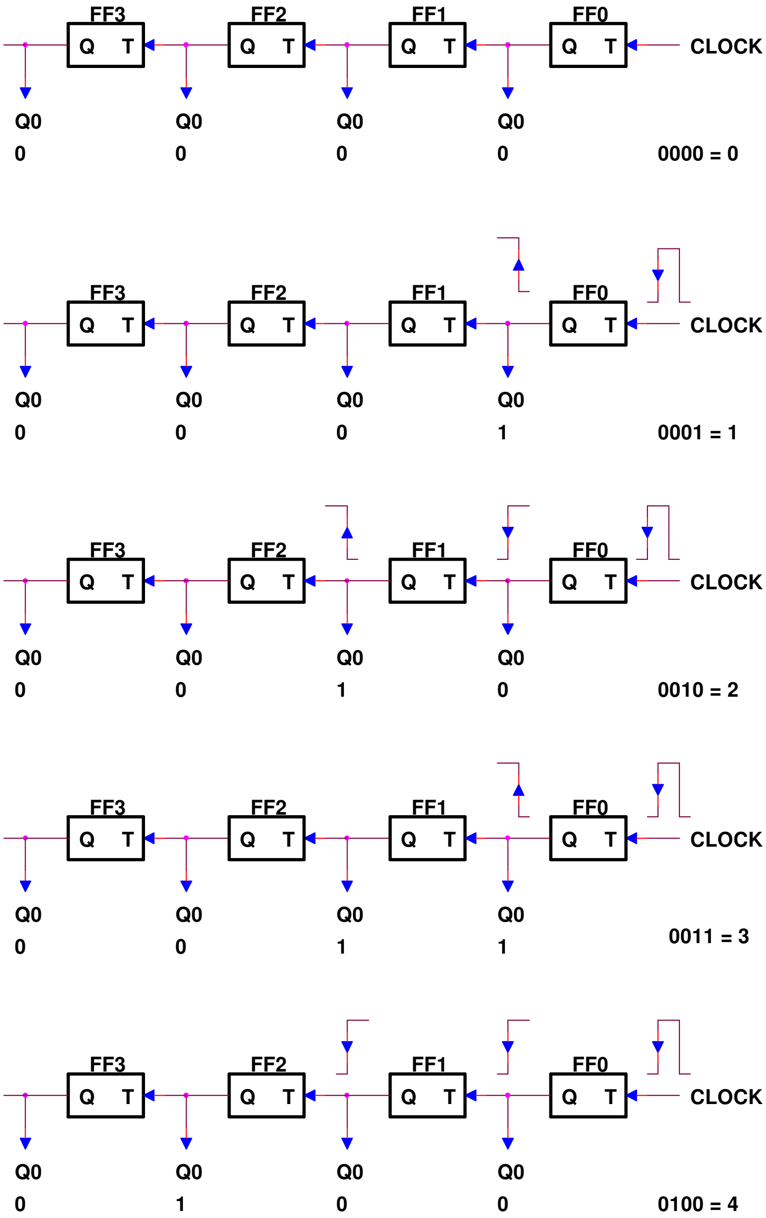
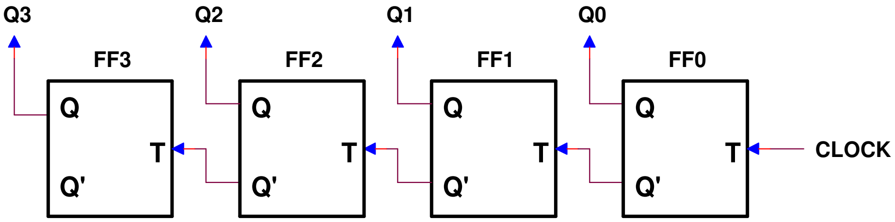


Fig. 8-14 : Up-Counter Hardware Sequence

8-4



c. "DOWN" Counter

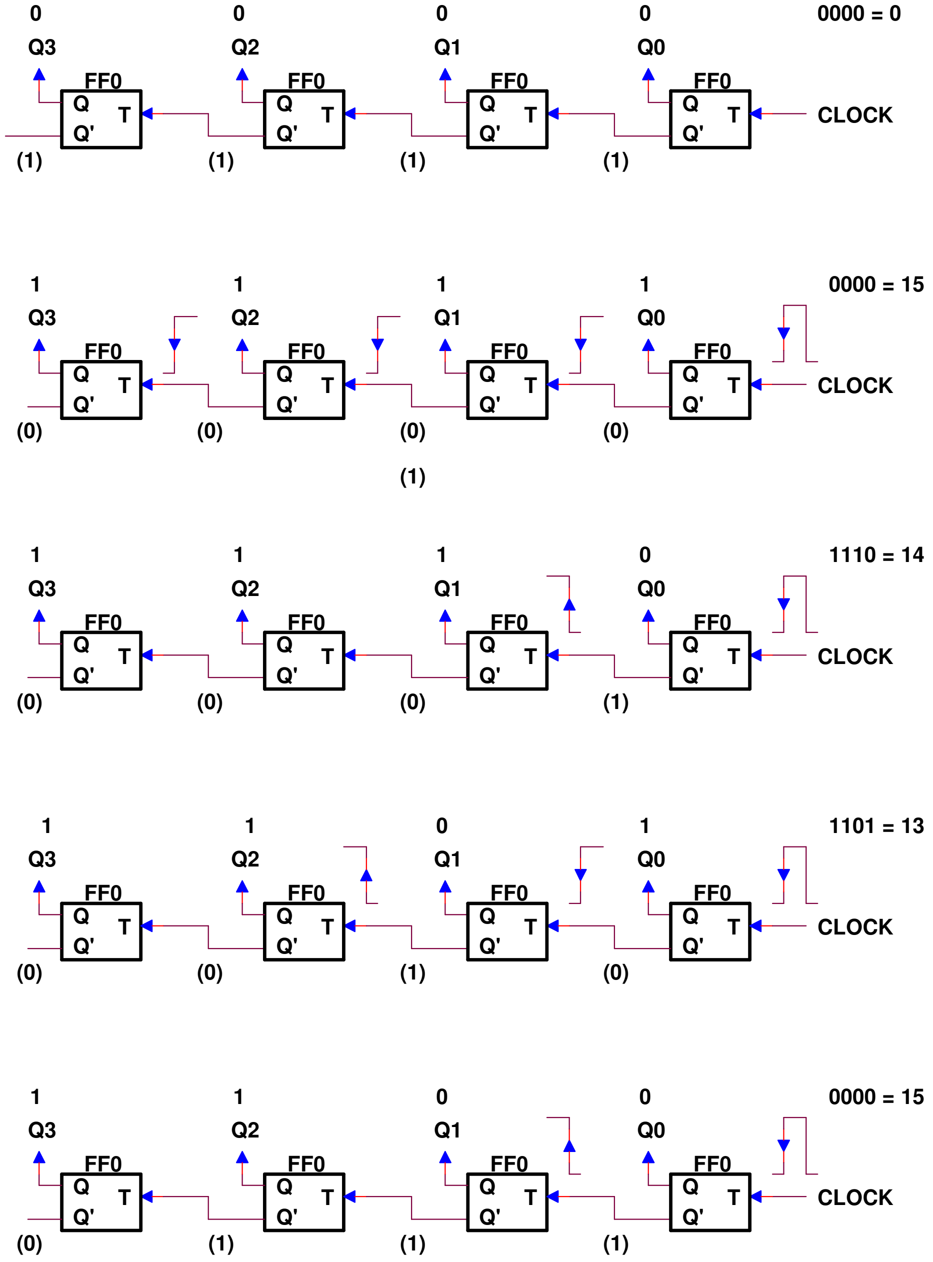
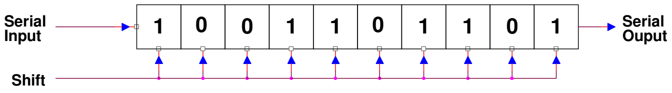
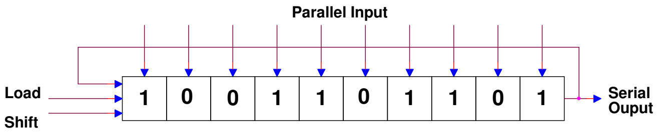


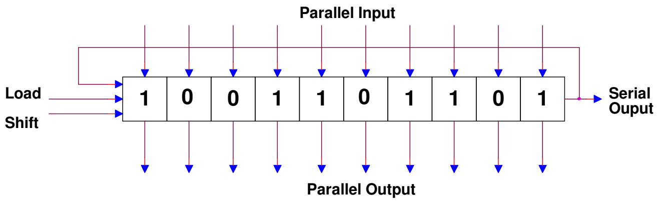
Fig. 8-15 : Down-Counter Hardware Sequence



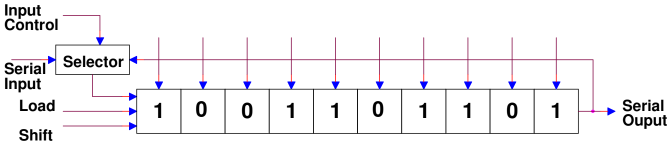
a. Shift Register - Serial In,Serial Out



b. Shift Register (Cyclic) - Parallel In,Serial Out

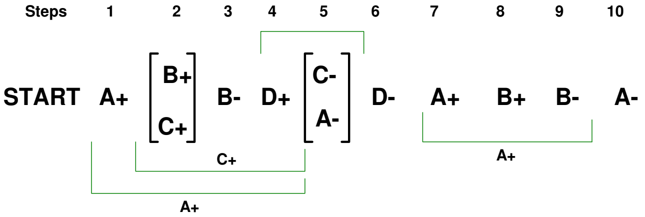


c. Shift Register (Cyclic) - Parallel In,Serial/Parralel Out

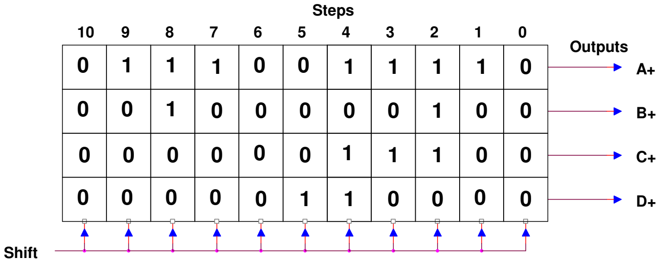


d. Shift Register (Cyclic) - Serial/Parallel In,Serial Out

Fig. 8-16 : 10 Stages Shift Registers



a. 10-Step Synchronous Sequence Definition

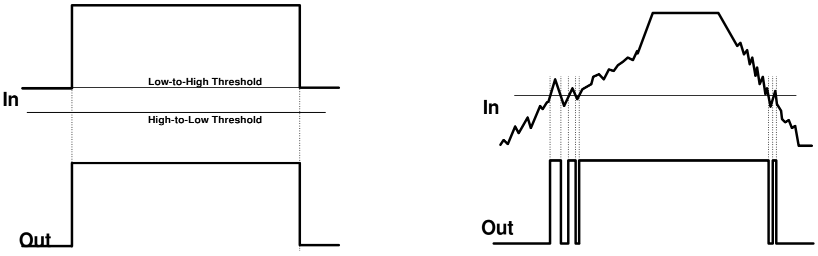


10	9	8	7	6	5	4	3	2	1	0	Outputs
0	1	1	1	0	0	1	1	1	1	0	A+
0	0	1	0	0	0	0	0	1	0	0	B+
0	0	0	0	0	0	1	1	1	0	0	C+
0	0	0	0	0	1	1	0	0	0	0	D+

b. Shift Register With Four Parallel Tracks Implementation

Fig. 8-17 : Implementation of Multi Track Shift Register for Operating Synchronous Sequence

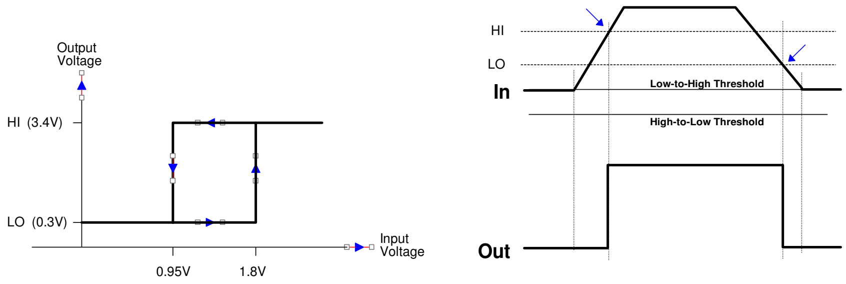
a. Typical Gate I/O



b. Correct Input Signal

c. Noisy Input Signal

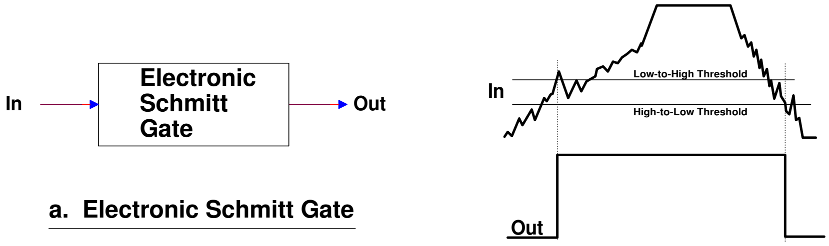
Fig. 8-18 : Response of Standard Gate



a. Hysteresis I/O Graph

b. Typical Response

Fig. 8-19: Schmitt-Trigger Response



a. Electronic Schmitt Gate

b. Response of Schmitt Trigger Gate

Fig. 8-20: Standard/Schmitt Response to Fuzzy Waveform

8-7

Decimal Value	Natural Binary Code	Reflected Cyclic Code
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

Fig. 8-21: Comparison Between Natural-Binary and Reflected-Cyclic Code

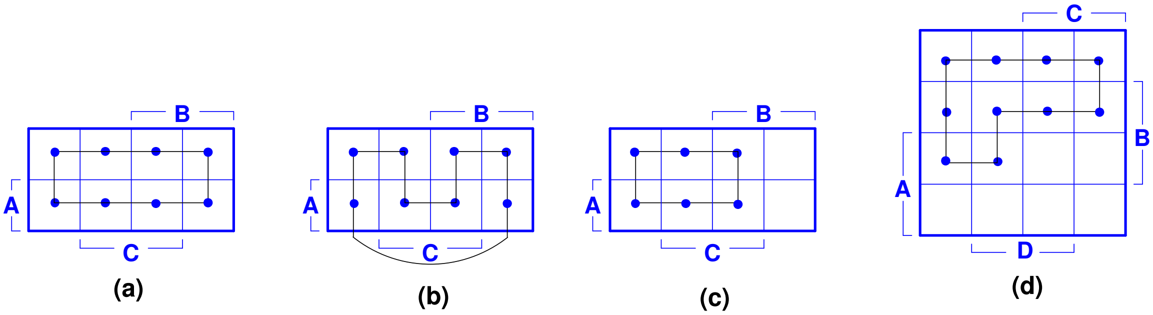
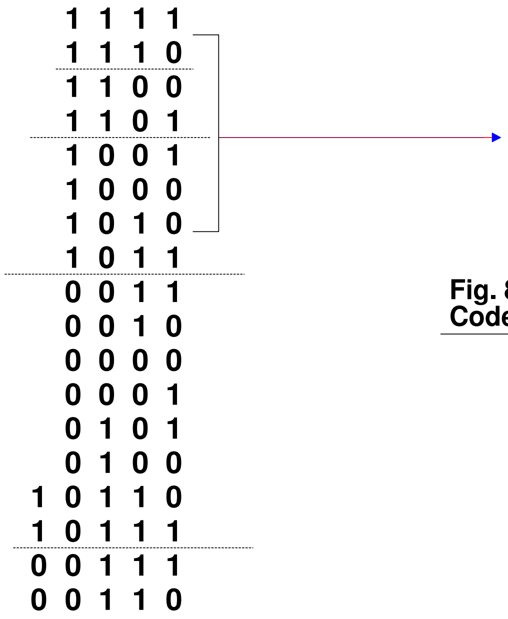


Fig. 8-22: Use of Karnaugh Map to Obtain Cyclic Codes

1 1 0
1 0 0
1 0 1
0 0 1
0 0 0
0 1 0

Fig. 8-23: Reflected Cyclic Code for 6 Numbers



1 1 1 1
1 1 1 0
1 1 0 0
1 1 0 1
1 0 0 1
1 0 0 0
1 0 1 0
1 0 1 1
0 0 1 1
0 0 1 0
0 0 0 0
0 0 0 1
0 1 0 1
0 1 0 0
1 0 1 1 0
1 0 1 1 1
0 0 1 1 1
0 0 1 1 0

Fig. 8-24: Construction of "Reflected" Cyclic Code

Weight Number	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

a. Natural Code
(No Parity)

P = Parity Bit

Weight Number	8	4	2	1	P
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

b. Natural Code With Binary Parity Bit
(Even Parity)

Fig. 8-25: Truth Tables for Natural Code

Weight Number	8	4	2	1	P
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0

a. Weighted BCD Code

Weight Number	7	4	2	1	P
0	1	1	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	1	0	0	0	1
8	1	0	0	1	0
9	1	0	1	0	0

b. 2-out-of-5 BCD Code

Fig. 8-26: Truth Tables for BCD Code
(With Even Parity bit)

Error Detection Not Available

a. Natural Code

Detect ODD numbers of Errors

b. Single Parity Bit

Correct Single Error &
Detect 2 Errors

c. Multiple Parity Bits

Fig 8-27 : Common Modes

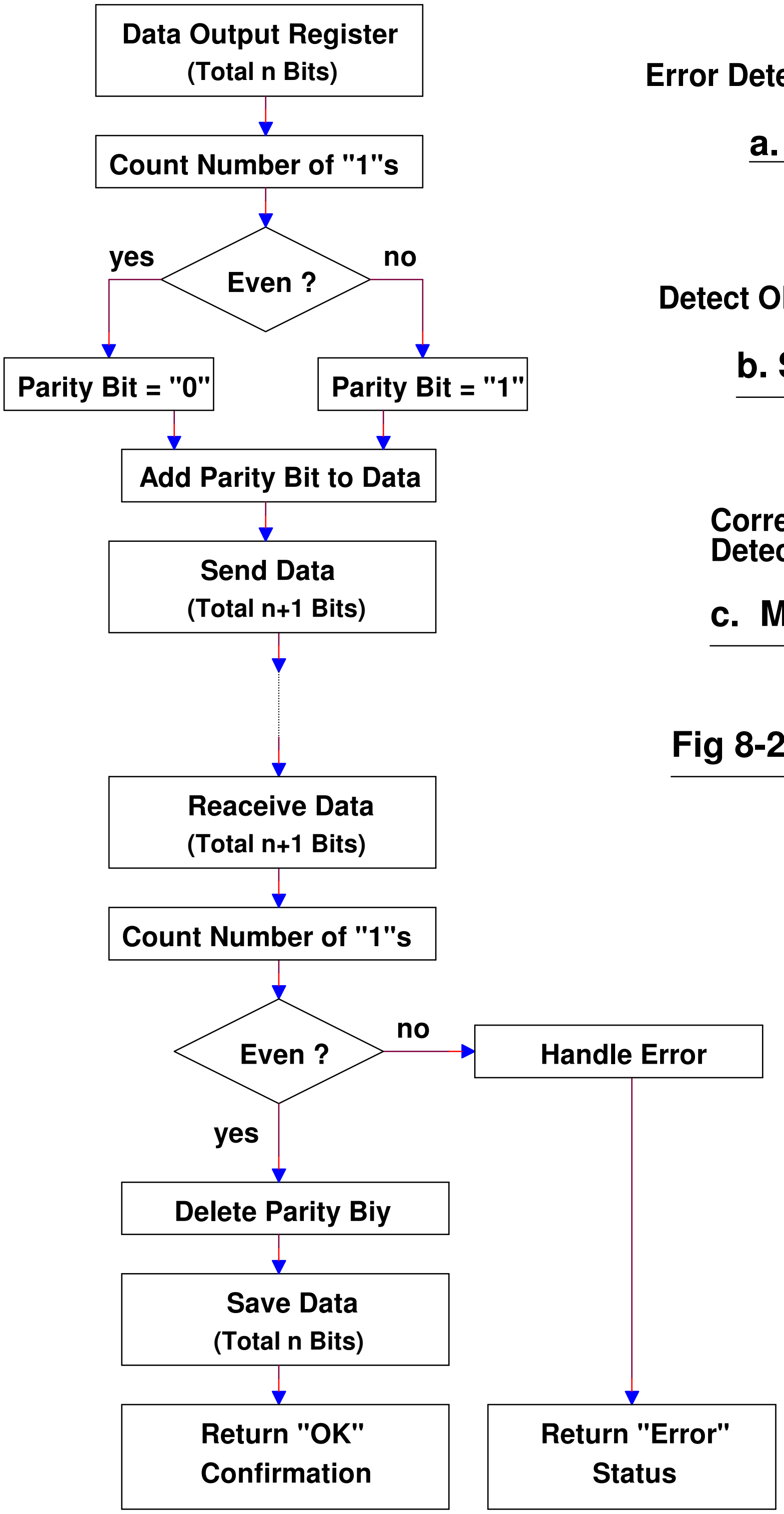
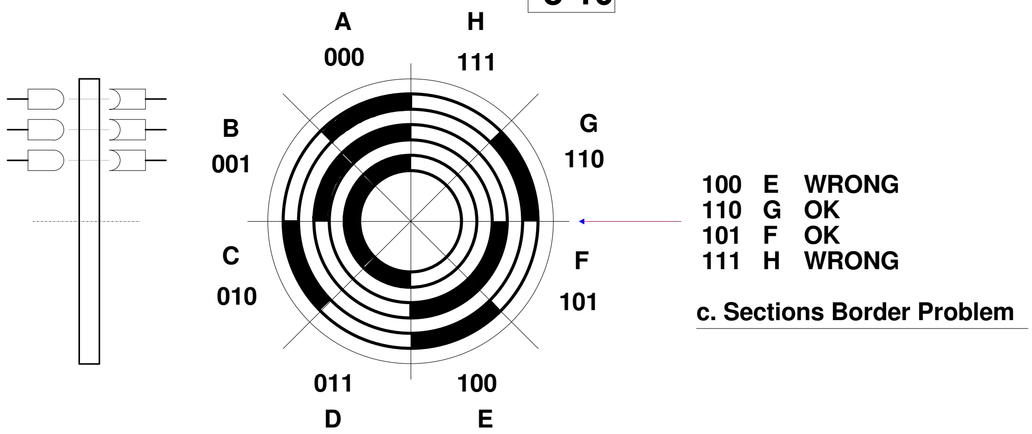


Fig 8-28: Parity Check Process



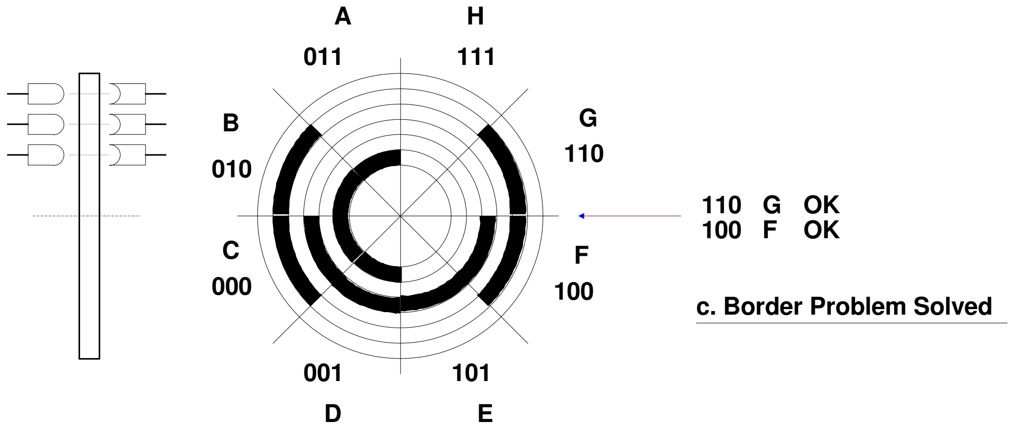
c. Sections Border Problem

b. Encoder Photo-Cells

a. Encoder Plate

Fig. 8-29: Binary 8-Sections Encoder

Absolute Coordinates



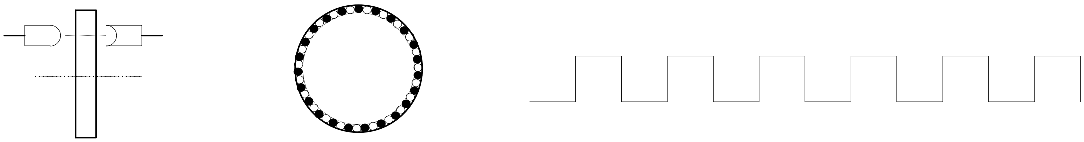
c. Border Problem Solved

b. Encoder Photo-Cells

a. Encoder Plate

Fig. 8-30: Typical 8-Sections Encoder

Absolute Coordinates



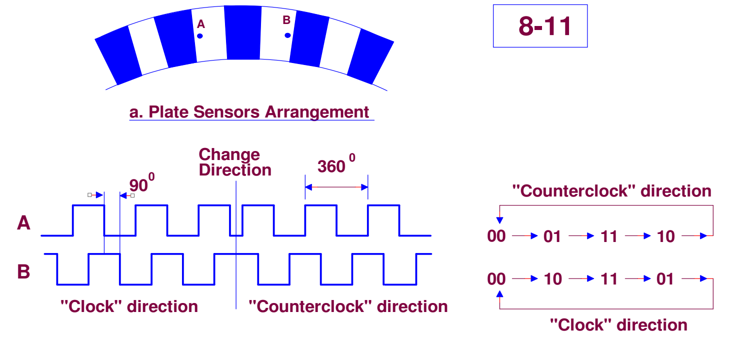
b. Encoder Photo-Cells

a. Encoder Plate

c. Incremental Encoder Output

No Sense for Direction

Fig. 8-31:Single Output Incremental Encoder

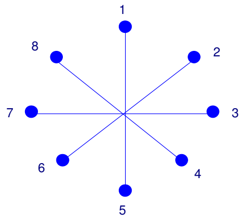


a. Plate Sensors Arrangement

b. Encoder Code Waveforms for Both Directions

c. Cells A and B sequences

Fig. 8-32: Two Output Incremental Encoder

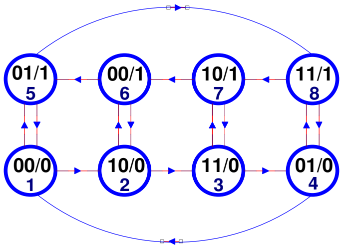


d. Merge Diagram

AB	00	01	11	10
CR1	①$_0$	⑤$_1$	8	2
CR2	⑥$_1$	5	3	②$_0$
CR3	6	4	③$_0$	⑦$_1$
CR4	1	④$_0$	⑧$_1$	7

e. Merged Flow-table & State Assignment

a. Control System Block Diagram



b. Primitive Flow-diagram

AB	00	01	11	10
CR1	①	⑤	8	2
CR2	⑥	5	3	②
CR3	6	4	③	⑦
CR4	1	④	⑧	7

f. Merged Flow-Table States Transitions

AB	00	01	11	10
CR1	O	1	(1)	(0)
CR2	1	(1)	(0)	O
CR3	(1)	(0)	O	1
CR4	(0)	O	1	(1)

D

g. Output Table

AB 00	01	11	10	D
①	5	-	2	O
6	-	3	②	O
-	4	③	7	O
1	④	8	-	O
1	⑤	8	-	1
⑥	5	-	2	1
6	-	3	⑦	1
-	4	⑧	7	1

c. Primitive Flow-table

~~S1 = CR4.A.B' + CR2.A'B'~~
S2 = CR1.AB' + CR3.A.'B'
S3 = CR2.A.B + CR4.A.B'
S4 = CR1.A.B + CR3.A'B

R1 = CR2.A.B' + CR4.A.B
R2 = CR1.A'B + CR3.AB
R3 = CR2.A'.B' + CR4.A'B
R4 = CR1.A'.B' + CR3.AB'

D = CR1.B + CR2.A' + CR3.B' + CR4.A

h. Exitation and Output Functions

Initially, A=B=0 and all Flip-flops are in reset state (CRi=0).

At that state, Flip-flop 1 must be set, with no confilict later, that is to say, while CR1 and CR2 are not set.

S1 = A'B'.CR2'.CR3'

i. Corrected System Initialization

Fig. 8-33: Motor Direction Detection System Design (Copy of 6-4)

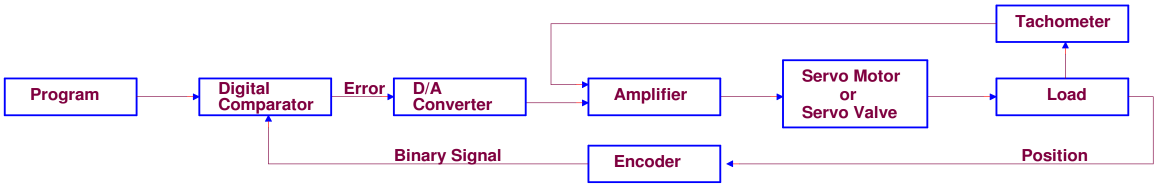


Fig. 8-34: Simplified Block Diagram of a Closed Loop NC System (One Control Axis)

Fig. 8-35: Block Diagram of an Open-Loop NC System (One Control Axis)

CHAPTER 9

ITERATIVE SYSTEMS

Iterative Binary Adder
Iterative Design Samples
Binary Numbers Comparator
Cyclic Codes Converters

9-1

X1	X0	Y1	Y0	Z2	Z1	Z0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

a. Truth Table

Z2 = X1.Y1 + X1.X0.Y0 + X0.Y1.Y0

Z1 = X1'X0.Y1'Y0 + X1.X0.Y1.Y0 + X1.Y1'Y0' +
+ X1.X0'Y1' + X1'Y1.Y0' + X1'X0'Y1

Z0 = X0.Y0' + X0'Y0

b. Sum Output Minimized Functions

* 3-Bit Numbers Adder requires 6-Variable Map
* 4-Bit Numbers Adder requires 8-Variable Map
* 32-Bit Numbers Adder requires 64-Variable Map
* Karnaugh Map Metode Is Limited to 4-Bit Numbers

c. Karnaugh Map Methose Limitation

Fig. 9-1 : Adder for 2-Bit Numbers

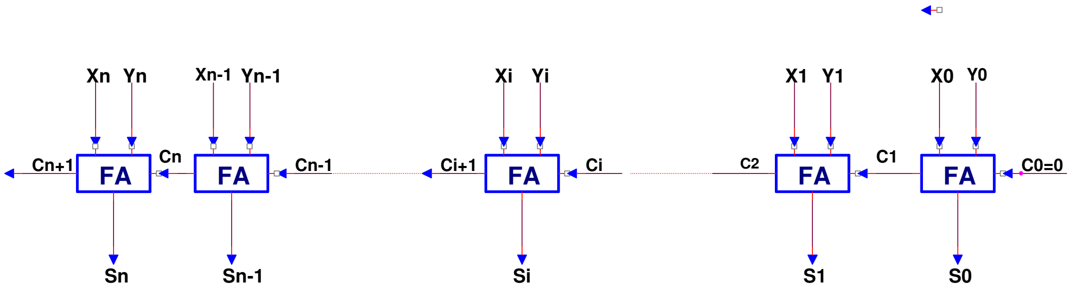
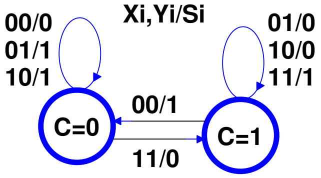


Fig. 9-2 : Iterative Adder Block Diagram



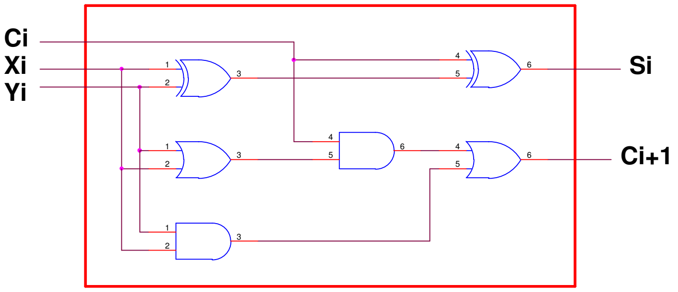
a. General Cell (FA) Flow Chart

Ci	Xi	Yi	Ci+1	Si
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Si = (XI) xor (Yi) xor (Ci)

Ci+1 = Xi.Yi+ Ci (Xi + Yi)

b. FA Truth Table



c. FA Logic Circuit

Fig. 9-3 : Iterative Binary Adder Design

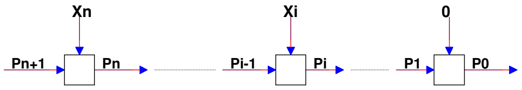
DESIGN OF A PARITY CHECKER

Checker detects if number of "1" in a binary vector, is odd or even.
It produces a single output line P, where :

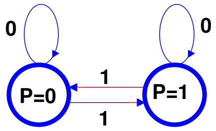
P=1 denotes odd parity
P=0 denotes even parity

a. Requirements Definition

b. Simplified Block Diagram



c. Iterative Block Diagram



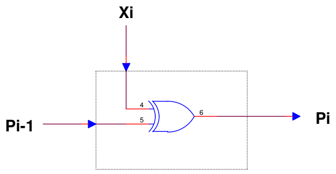
d. General Cell Flow Chart

Pi-1	Xi	Pi
0	0	0
0	1	1
1	0	1
1	1	0

e. General Cell Truth Table

Pi = Pi-1.Xi' + Pi-1'.Xi = Pi-1 xor X

f. General Cell Function



g. General Cell Logic Circuit

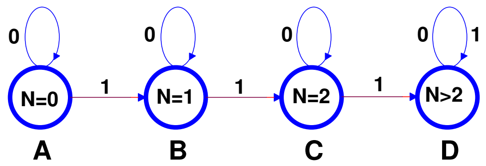
Fig. 9-4: Iterative Parity Checker

DESIGN OF 2-OUT-OF-5 CHECKER

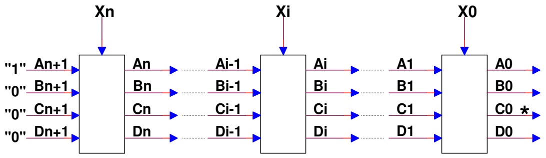
A specific code consists of vectors that contain exactly two "1"'s, and all other bits are "0". (11000000, 00101000, 010000001 etc).
Design a circuit that checks vectors and announces Z=1 when detected vector been found valid.

a. Requirements Definition

b. Simplified Block Diagram



c. General Cell Flow Chart



d. Iteratice Block Diagram

Ai-1	Bi-1	Ci-1	Di-1	Xi	Ai	Bi	Ci	Di
1	-	-	-	0	1	0	0	0
1	-	-	-	1	0	1	0	0
-	1	-	-	0	0	1	0	0
-	1	-	-	1	0	0	1	0
-	-	1	-	0	0	0	1	0
-	-	1	-	1	0	0	0	1
-	-	-	1	0	0	0	0	1
-	-	-	1	1	0	0	0	1

e. General Cell Truth Table

Ai = Ai-1.Xi'
Bi = Ai-1.Xi + Bi-1.Xi'
Ci =Bi-1.Xi + Ci-1.Xi'
Di =Ci-1.Xi +Di-1

f. General Cell Function

Fig. 9-5 : Iterative 2-Out-of-n Checker

DESIGN OF A BINARY COMPARATOR

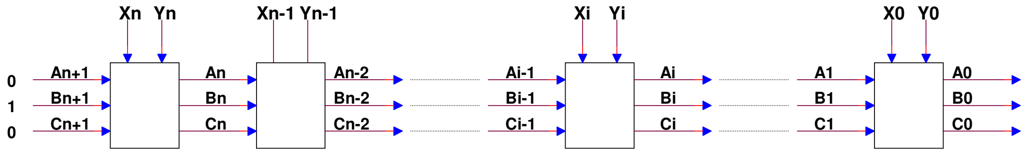
Comparator compares 2 binary numbers X,Y (of equal length), and produces 3 output signals:

Line A : A=1 if X>Y
Line B : B=1 if X=Y
Line C : C=1 if X<Y

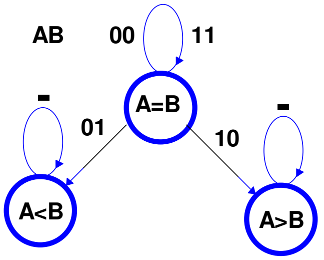
Design refers to positive numbers of n+1 bits, where n may vary according to case.

a. Requirements Definition

b. Simplified Block Diagram



c. Iterative Block Diagram



d. General Cell Flow Chart

$A_i = A_{i-1} + B_{i-1}.X_i.Y_i'$

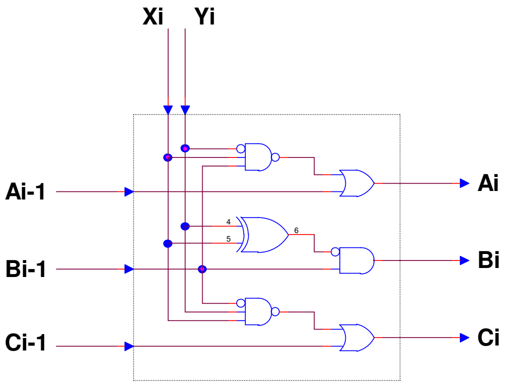
$B_i = B_{i-1}(X_i.Y_i+X_i'.Y_i') = B_{i-1}(X_i \text{ xor } Y_i)'$

$C_i = C_{i-1} + B_{i-1}.X_i.'Y_i$

f. General Cell Functions

Ai-1	Bi-1	Ci-1	Xi	Yi	Ai	Bi	Ci
1	0	0	-	-	1	0	0
0	0	1	-	-	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	1	0	0	1
0	1	0	1	0	1	0	0
0	1	0	1	1	0	1	0

e. General Cell Truth Table



g. General Cell Logic Circuit

Fig. 9-6 : Iterative Binary Numbers Comparator

Decimal Value	Natural Binary Code	Reflected Cyclic Code
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

Fig. 9-7 : Comparison Between Natural-Binary and Reflected-Cyclic Code

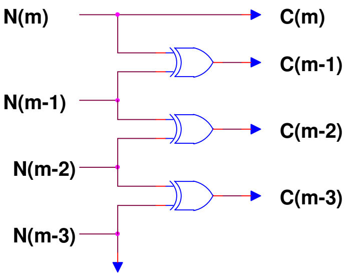


Fig. 9-8 : Iterative Circuit for Translating Natural-Binary into Reflected-Cyclic Code

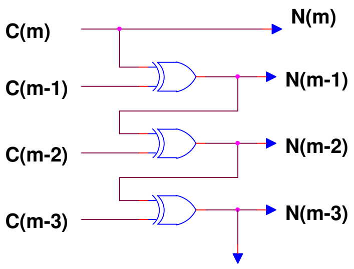


Fig. 9-9 : Iterative Circuit for Translating Reflected-Cyclic into Natural-Binary Code

N(i)	N(i-1)	C(i-1)
0	0	0
0	1	1
1	0	1
1	1	0

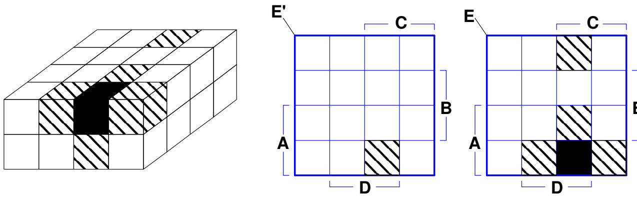
$$C(i-1) = N(i) \oplus N(i-1)$$
$$N(i-1) = N(i) \oplus C(i-1)$$

Fig. 9-10 : Truth Table for Translating Natural-Binary into Reflected-Cyclic Code

CHAPTER 10

MULTI VARIABLES KARNAUGHT MAPS

5-Variables Map
6-Variables Map
7-Variables Map
8-Variables Map



a. 3-D Representation

b. Flat Representation

Fig. 10-1 : Three-Dimentional Five-Variable Karnaugh Map, and Its Developmenet

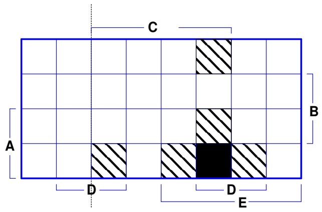
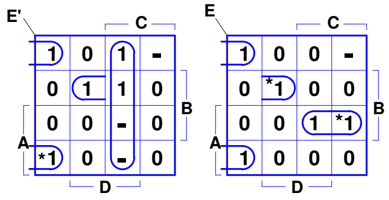


Fig. 10-2 : Optional Construction of Five-Variable Karnaugh Map (Not Recommended)



1	0	1	-
0	1	1	0
0	0	-	0
*1	0	-	0

1	0	0	-
0	*1	0	0
0	0	1	*1
1	0	0	0

b. Karnaugh Map

*

T=A'B'C'D'+A'BDE'+A'B'CDE'+AB'C'D'+A'BC'DE+ABCDE+ABCD'E

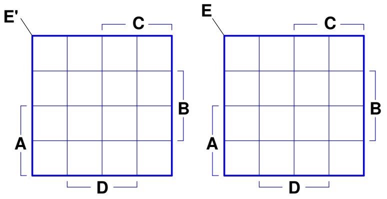
with impossible conditions : A'B'CD'=1 ACDE'=1

a. Original Function

T=B'C'D'+A'BC'D+CDE'+ABCE

c. Simplified Function

Fig. 10-3 : 5-Variable Minimzation Exercise (With Solution)



b. Karnaugh Map

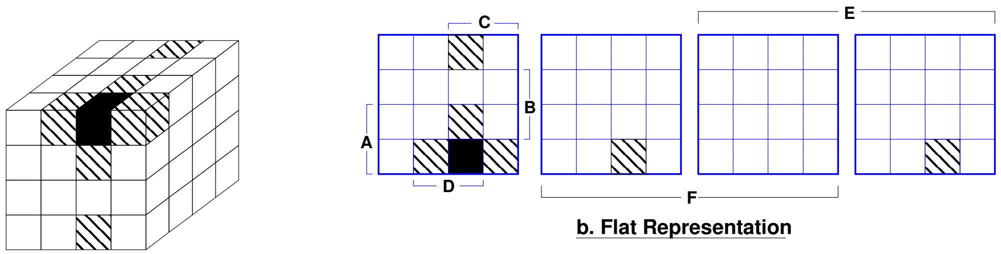
T=A'B'C'E'+A'B'CD'+ABC'DE'+AB'C'E'+AB'CD'E'+B'C'D'E+A'B'C'DE+CD'E

a. Original Function

T=

c. Simplified Function

Fig. 10-4 : 5-Variable Minimzation Exercise (Without Solution)

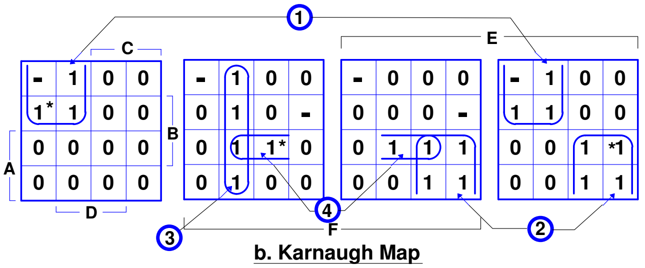


a. 3-D Representation

b. Flat Representation

Fig. 10-5 : Three-Dimentional slx-Variable Karnaugh Map, and Its Developmenet

---



-	1	0	0	-	1	0	0	-	0	0	0	-	1	0	0
1*	1	0	0	0	1	0	-	0	0	0	-	1	1	0	0
0	0	0	0	0	1	1*	0	0	1	1	1	0	0	1	*1
0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1

b. Karnaugh Map

T=A'BC'F'+AB'C'DE'F+ABDF+ACDE+ACD'E+
+A'C'DE'+A'C'DEF'

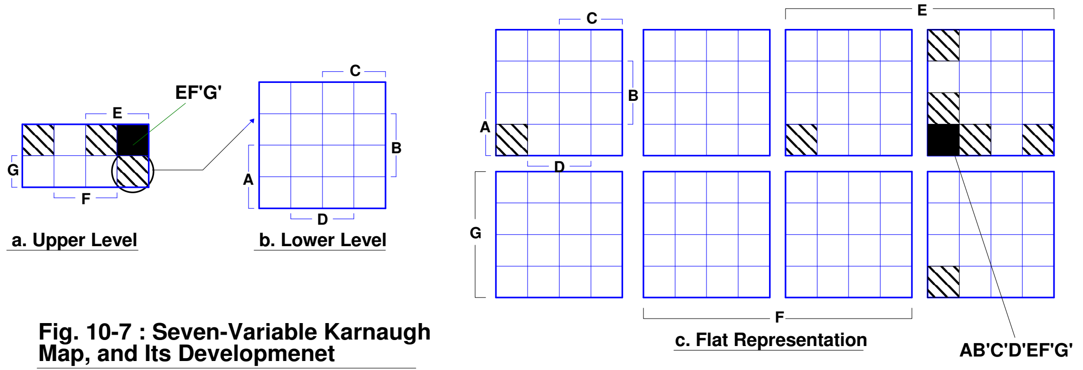
with impossible conditions :
A'B'C'D'=1 A'BCD'F=1

a. Original Function

T=A'C'F'+ACE+C'DE'F+ABDF
① ② ③ ④

c. Simplified Function

Fig. 103-6 : 6-Variable Minimzation Exercise (With Solution)



a. Upper Level

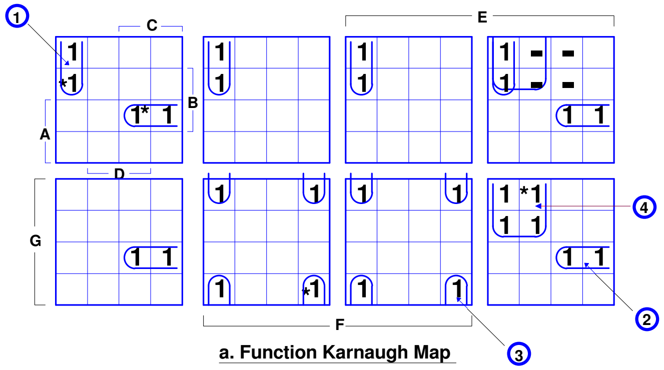
b. Lower Level

c. Flat Representation

Fig. 10-7 : Seven-Variable Karnaugh Map, and Its Developmenet

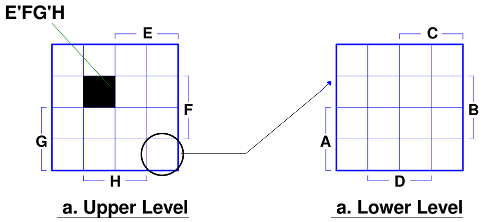
T=A'C'D'G'+ABCF'+B'D'FG+A'C'EF'
① ② ③ ④

b. Simplified Function



a. Function Karnaugh Map

Fig. 10-8 : 7-Variable Minimzation Exercise (With Solution)



a. Upper Level

a. Lower Level

Fig. 10-9 : Eight-Variable Karnaugh Map, and Its Developmenet

	Number of Variables in The Function						
Size of Cell	2	3	4	5	6	7	8
Single Square	2	3	4	5	6	7	8
2-Square Cell	1	2	3	4	5	6	7
4-Square Cell		1	2	3	4	5	6
8-Square Cell			1	2	3	4	5
16-Square Cell				1	2	3	4
32-Square Cell					1	2	3
64-Square Cell						1	2
128-Square Cell							1

Fig. 10-11 : Number of Variables Required to Define Certain Cell Address

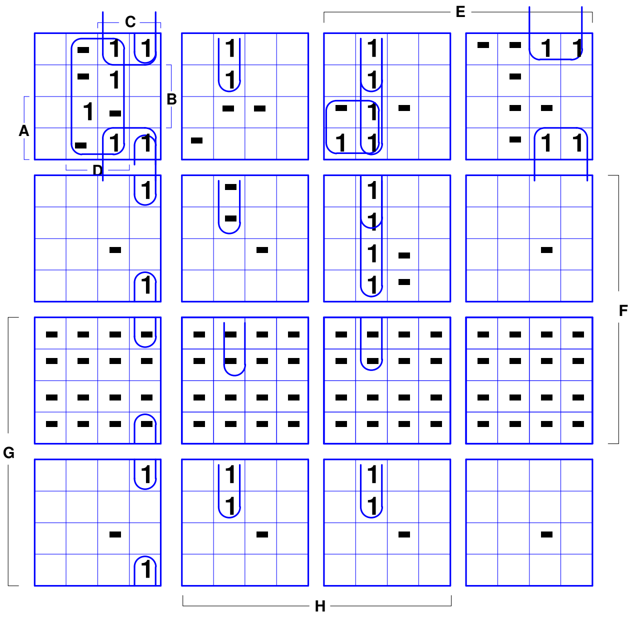


Fig. 10-10 : 8-Variable Minimzation Exercise

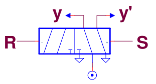
10-3

Semi-Karnaugh Maps

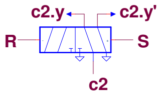
T.B.D.

CHAPTER 11

PNEUMATIC FLOW-TABLE METHODE



a. Flip-Flop Valve



b. Steering Valve

Fig.11-1 : Flip-Flop Valve and Steering Valve

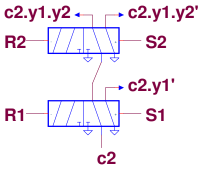


Fig. 11-2 : Two Steering Valves providing 3 addresses

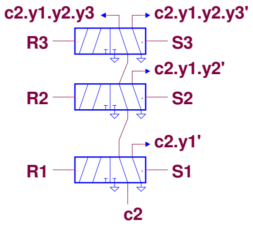


Fig. 11-3 : Three Steering Valves providing 4 addresses

START,A+,A-,A+,A-,A+,A-

a. Process sequence

a1	a2	A+	A-
①		1	
	②		1
③		1	
	④		1
⑤		1	
	⑥		1

b. Flow Table

a1	a2	A+	A-
y1' ① R3		1	
	y3' ② S1 R2		1
y1.y2' ③ S3 R4		1	
	y3.y4' ④ S2		1
y1.y2 ⑤ S4		1	
	y3.y4 ⑥ R1		1

c. Complete Flow Table

S1= a2.y3'
R1 = a2.y3.y4

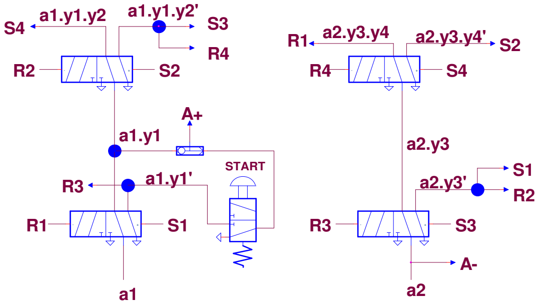
S2 = a2.y3.y4'
R2 = a2.y3'

S3 = a1.y1.y2'
R3 = a1.y1'

S4 = a1.y1.y2
R4 = a1.y1.y2'

A+ = a1.y1'.Start+a1.y1.y2'+a1.y1.y2 = a1.y1'.Start+a1.y1
A- = a2.y3'+a2.y3.y4'+a2.y3.y4 = a2.y3'+a2.y3 = a2

d. Exitation and Output Functions



e. Control Circuit

Fig. 11-4 : Pneumatic Flow-Table Design for sequences START,A+,A-,A+,A-,A+,A-

---

START,A+,B+,A-,B-, $\left|\begin{matrix}A+\\B+\end{matrix}\right|$,A-,B-

a. Process sequence

a1.b1	a2.b1	a2.b2	a1.b2	A+	A-	B+	B-
y1'①				1			
	① R2					1	
		y2' ③ R3			1		
			y3' ④ S1				1
y1 ⑤ s2				1		1	
		y2 ⑥ S3			1		
			y3 ⑦ R1				1

b. Complete Flow Table

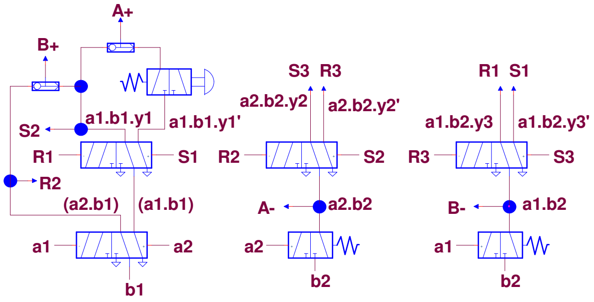
S1 = a1.b2.y3'
R1 = a1.b2.y3

S2 = a1.b1.y1
R2 = a2.b1

S3 = a2.b2.y2
R3 = a2.b2.y2'

A+ = a1.b1.y1'.Start+a1.b1.y1 = a1.b1.Start+a1.b1.y1
A- = a2.b2.y2'+a2.b2.y2 = a2.b2
B+ = a2.b1+a1.b1.y1
B- = a1.b2.y3'+a1.b2.y3 = a1.b2

c. Exitation and Output Functions



d. Control Circuit

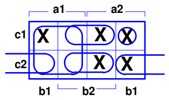
Fig. 11-5 : Pneumatic Flow-Table Design for Sequence START,A+,B+,A-,B-, $\left|\begin{matrix}A+\\B+\end{matrix}\right|$,A-,B-

START,A+,B+,B-,C+,B+,B-,C-,A-

a. Process sequence

a1.b1.c1 (a1)	a2.b1.c1	a2.b2.c1 (b2.c1)	a2.b1.c2 (b1.c2)	a2.b2.c2 (b2.c2)	A+	A-	B+	B-	C+	C-
① R1					1					
	y1' ②						1			
		③ S1 R2						1		
	y1.y2' ④ R3								1	
			y3' ⑤				1			
				⑥ S3				1		
			y3 ⑦ S2							1
	y1.y2 ⑧					1				

b. Complete Flow Table



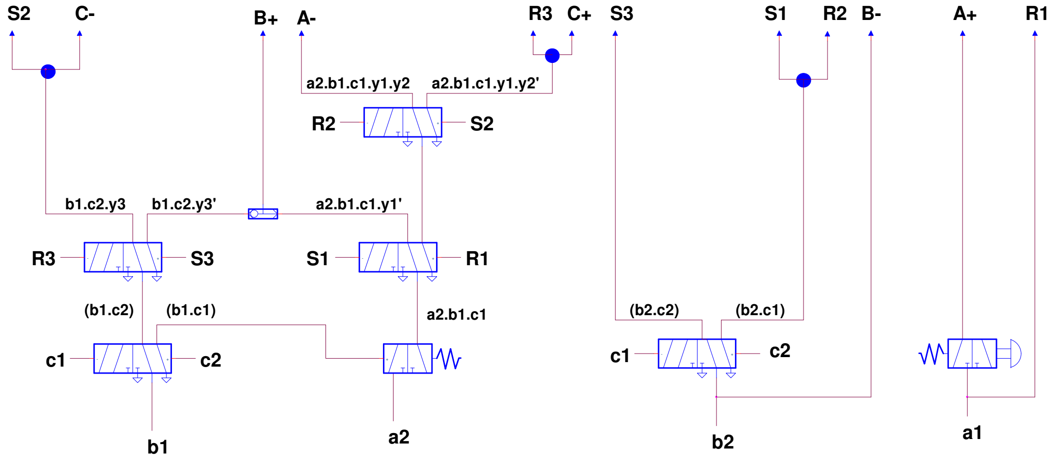
a1 b1 c1 → a1
a2 b2 c1 → b2 a1
a2 b1 c2 → b1 c2
a2 b2 c2 → b2 c2

c. Simplifiying Input Map

S1 = b2.c1
S2 = b1.c2.y3
S3 = b2.c2
R1 = a1
R2 = b2.c1
R3 = a2.b1.c1.y1.y2'

A+ = a1.START
A- = a2.b1.c1.y1.y2
B+ = a2.b1.c1.y1'+b1.c2.y3'
B- = b2.c1+b2.c2 = (b2)
C+ = a2.b1.c1.y1.y2'
C- = b1.c2.y3

d. Exitation and Output Functions



e. Control Circuit

Fig. 11-6 : Pneumatic Flow-Table Design for Sequence START,A+,B+,B-,C+,B+,B-,C-,A-

CHAPTER 12

MOTION ACTUATORS

Linear and Angular Motion
Electrical Linear Actuators
Electrical Rotary Actuators
Fluid-power Linear Actuators
Fluid-power Rotating Actuators
Boolean Functions Standard Formats

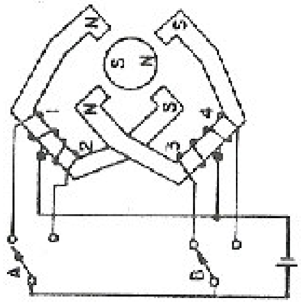


Fig. 11-3 : Eight types of
Pneumatic rotary actuators

hapter 1 : MOTION ACTUATORS

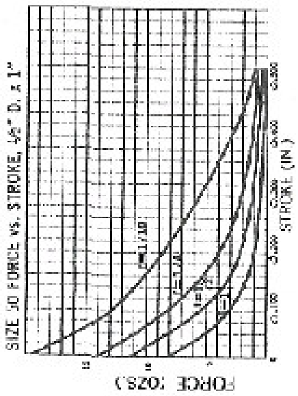


Fig. 11-2 : Force-stroke curve for
Typical ½" diameter solenoid

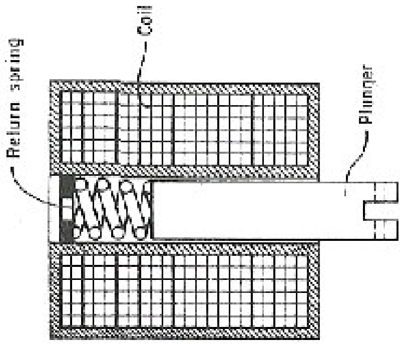


Fig. 11-1 : Solenoid

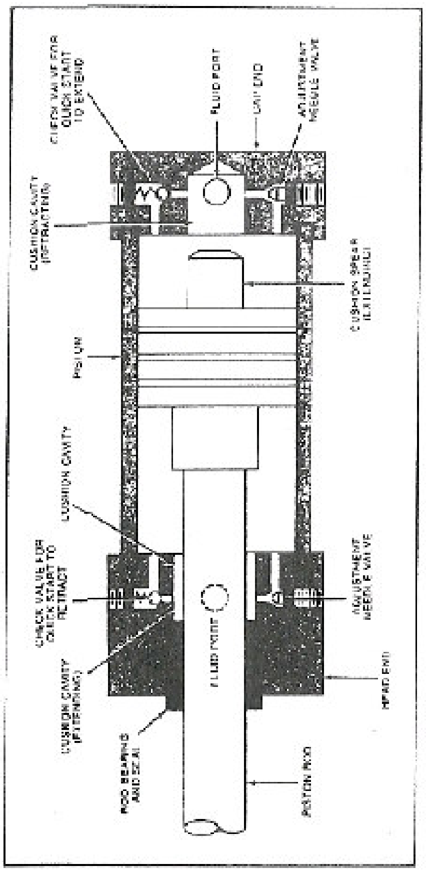


Fig. 11-4 : Pneumatic cylinder
With air cushions

12-2

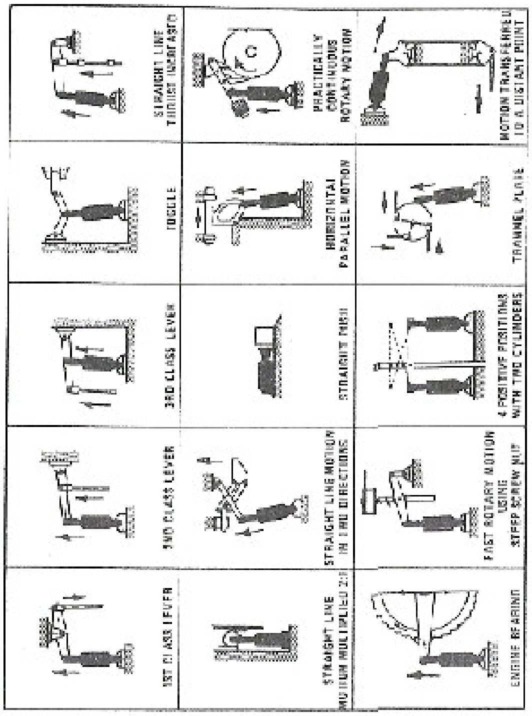
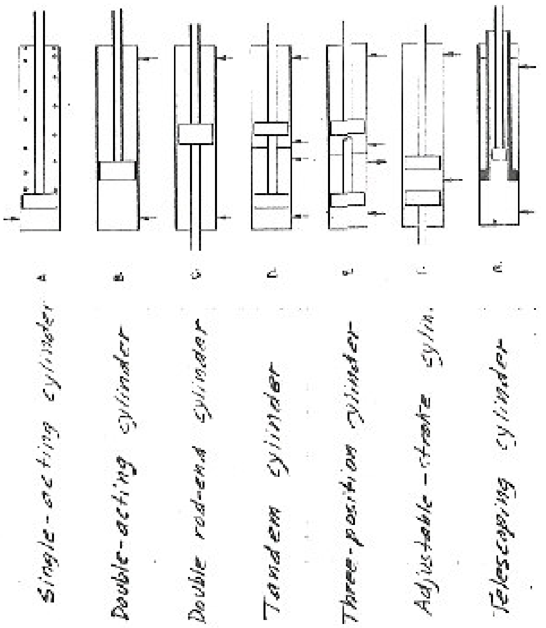


Fig. 12-6 : Fifteen ways of using cylindres



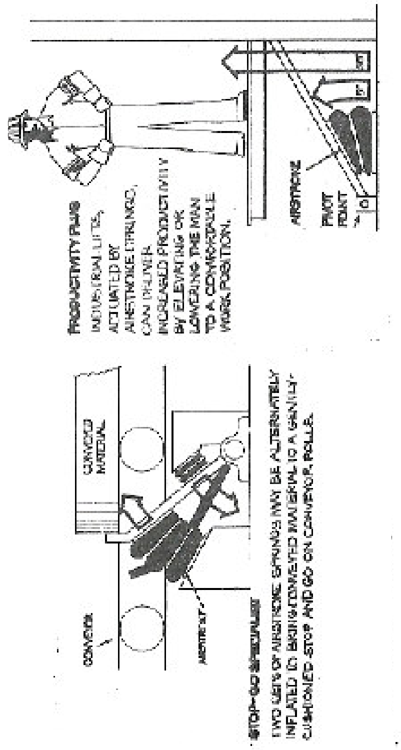


Fig. 12-7 : Two application of “airstroke” Actuator (Firestonr)

Fig. 12-5 : Basic cylinder types

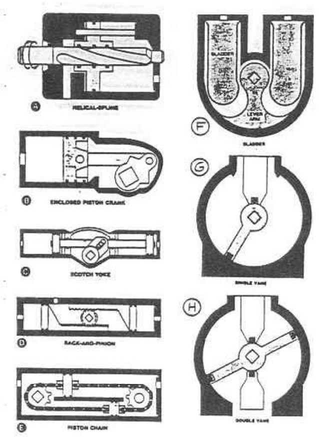


Fig. 12-8 : Eight types of
Pneumatic rotary actuators

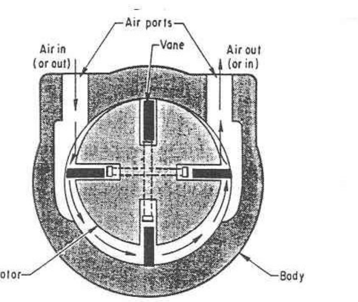


Fig. 3—Reversible vane-type air motor. To reverse direction of rotation, inlet line is moved from one port to the other.

Fig. 12-9 : Air motor

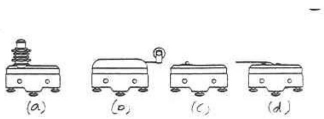
5

CHAPTER 13

SENSORS

Electric Position Sensors
Contacts Symbols
Photoelectric Sensors
Reed Switches
Proximity Sensors
Pressure, Flow and Level Switches
Pneumatic Limit Valves
Back-Pressure Sensors

13-1



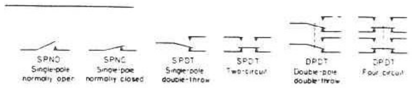


Fig. 13-1 : Common actuation methods for Limit Switches
(a) Overtravel plunger
(b) Lever roller
(c) Pin Plunger
(d) Lever

Fig. 13-2 : Common contact arrangements

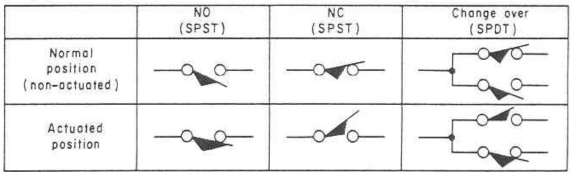


Fig. 13-3 : Limit-switch symbols for different contact configurations

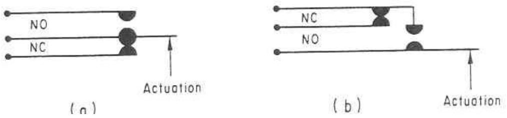


Fig. 13-4 : SPDT switch with (a) break-before-make (BBM) contacts, and (b) make-before-break (MBB) contacts

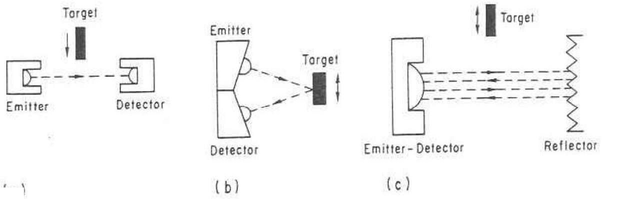


Fig. 13-5 : SPDT Three operating modes of photoelectric sensor –

(a) through beam
(b) reflection from target
(c) retroreflection

Fig. 13-11 : Use of magnetic proximity sensor
To measure rotational velocity

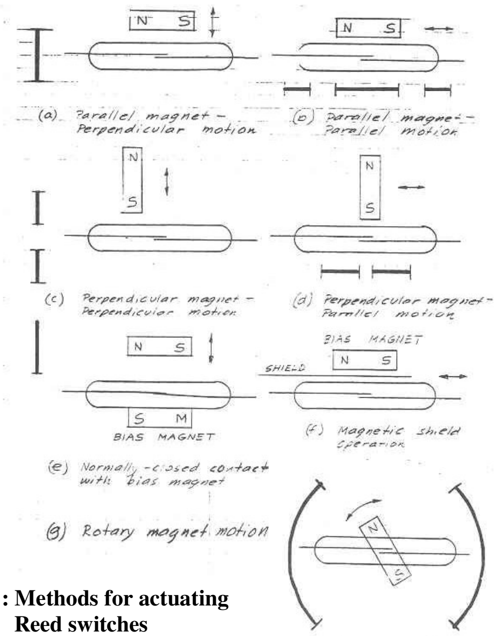
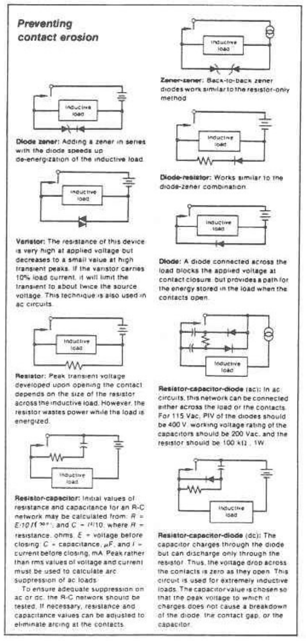


Fig. 13-6 : Methods for actuating Reed switches



Preventing contact erosion

Diode zener: Adding a zener in series with the diode speeds up de-energization of the inductive load.

Varistor: The resistance of this device is very high at applied voltage but decreases to a small value at high transient peaks. If the varistor carries 10% load current, it will limit the transient to about twice the source voltage. This technique is also used in ac circuits.

Resistor: Peak transient voltage developed upon opening the contact depends on the size of the resistor across the inductive load. However, the resistor wastes power while the load is energized.

Resistor-capacitor: Initial values of resistance and capacitance for an R-C network may be calculated from: $R = E/10I^{[illegible]}$; and $C = I^2/10$, where R = resistance, ohms, E = voltage before closing, C = capacitance, μF, and I = current before closing, mA. Peak rather than rms values of voltage and current must be used to calculate arc suppression of ac loads.

To ensure adequate suppression on ac or dc, the R-C network should be tested. If necessary, resistance and capacitance values can be adjusted to eliminate arcing at the contacts.

Zener-zener: Back-to-back zener diodes work similar to the resistor-only method.

Diode-resistor: Works similar to the diode-zener combination.

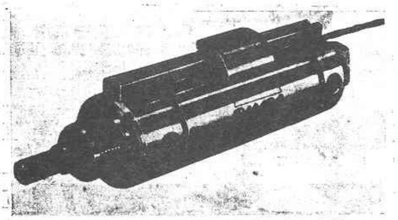
Diode: A diode connected across the load blocks the applied voltage at contact closure but provides a path for the energy stored in the load when the contacts open.

Resistor-capacitor-diode (ac): In ac circuits, this network can be connected either across the load or the contacts. For 115 Vac, PIV of the diodes should be 400 V, working voltage rating of the capacitors should be 200 Vac, and the resistor should be 100 kΩ, 1W.

Resistor-capacitor-diode (dc): The capacitor charges through the diode but can discharge only through the resistor. Thus, the voltage drop across the contacts is zero as they open. This circuit is used for extremely inductive loads. The capacitor value is chosen so that the peak voltage to which it charges does not cause a breakdown of the diode, the contact gap, or the capacitor.

Machine Design · May 13, 1982

Fig. 13-8 : Arc suppression method for protecting relay & switch controls

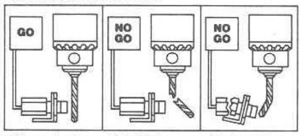


Air Cylinder Controls Magnetic Reed Switch

An SPST magnetic reed switch slides in a track on an air cylinder to operate relays, solenoids, timers, and other electrical equipment. A permanent magnet attached to the piston activates the switch. Six bore sizes are available from 1 1/16 to 2½ in. with strokes to 32 in.

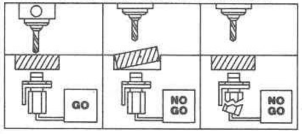
Fig. 13-7 : Use of reed switch to sense piston position

Broken Bit Detection —



With an unbroken bit, the Mark III output contacts are closed allowing regular operation.	If the bit is broken, the Mark III senses the damage and signals "No Go".	If a bent bit smashes the Mark III sensor, the machine is turned off.

Detection of Improper Part Position —



With the part in proper position, the Mark III output contacts are closed allowing regular operation.	With the part missing or in an irregular position, the Mark III signals "No Go".	If the sensor is broken or the cable is open or shorted, the Mark III turns off the machine.

Fig. 13-10 : Proximity sensor application

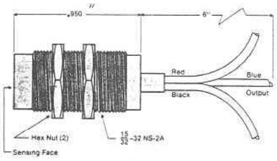


Fig. 13-9 : Proximity sensor

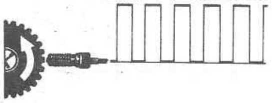


Fig. 2-11:
Use of magnetic proximity sensor to measure rotational velocity

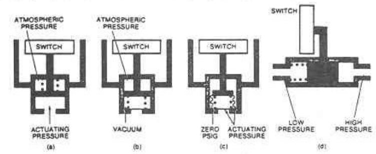


Fig. 1. Simplified sketches of three basic types of pressure switches: (a) gage senses pressure above atmosphere, (b) vacuum or compound (also gage) senses pressures less than atmospheric, (c) absolute senses pressures independent of atmospheric pressure, and (d) differential responds to a preset pressure difference.

Fig. 13-12 : Four basic types of pressure switches

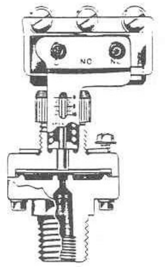


Fig. 13-13 : pressure switch (diagram and Step acts on a limit

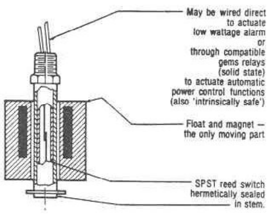


Fig. 13-14 : Level switch (based On reed switch)

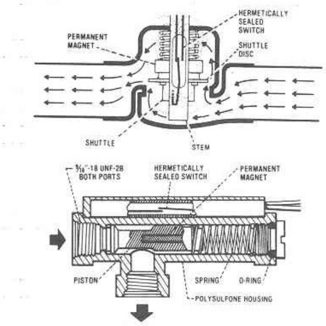


Fig. 13-15 : Two flow switches (based on reed switch)

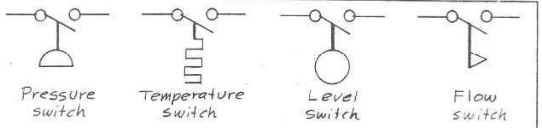


Fig. 13-16 : Symbols for various sensor switches (normally-open contacts)

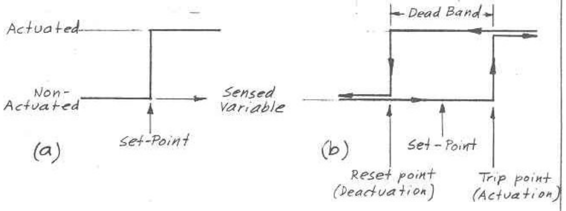


Fig. 13-17 : Sensor switch -
(a) without dead band
(b) With dead band (hysteresis)

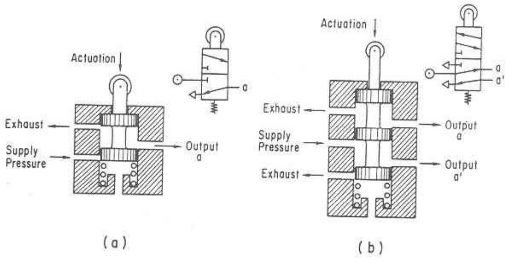


Fig. 13-18 :

Pneumatic limit valves and their fluid-power symbols:

(a) a 3/2 valve, and (b) a 5/2 valve.
(Spool Type)

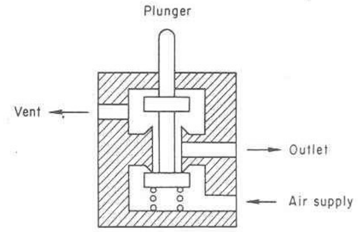


Fig. 13-19 :

3/2 limit valve
(Poppet Type)

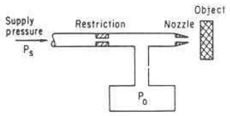


Fig. 13-20 : Flapper-Nozzle system used as back-pressure

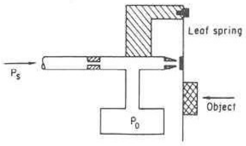


Fig. 13-21 : Back-pressure sensor with overtravel rotection

Fig. 13-21 : Back-pressure sensor with overtravel rotection

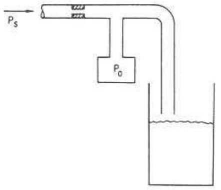


Fig. 13-22 : Detecting liquid level of power level using back-pressure sensor (bubble tube)

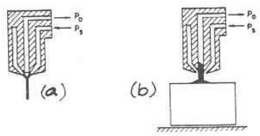


Fig. 13-23 : Annular back-pressure (a) output pressure P0 low

(b) sensor P0 high

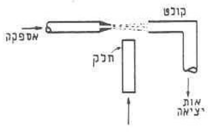


Fig. 13-24 : Interruptable-jet sensor

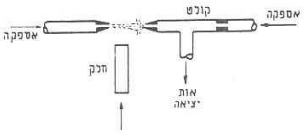


Fig. 13-25 : Interruptable-jet sensor (insensitive to dirt)

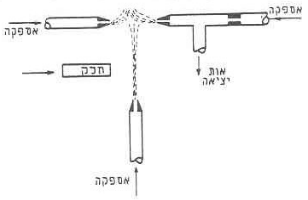
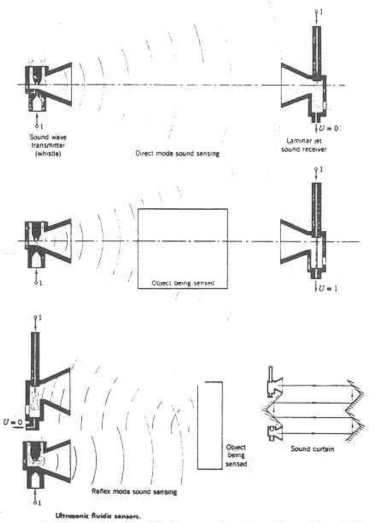


Fig. 13-26 : Interruptable-jet sensor (with remote actuation)



Ultrasonic fluidic sensors.

As illustrated in Fig. 4-12 the properties of sound waves extends the usefullness of the ultrasonic-type sensor: the reflex method of sensing can detect objects on a conveyor line or a sound curtain can detect the presence of a person entering a dangerous area.

Specifications

SUPPLY PRESSURE		0 TO 30 psig
DEFLECTION ANGLE		6° MINIMUM AND 45° MAXIMUM
DEFLECTION FORCE		APPROXIMATELY 1 OUNCE AT END OF ROD
MINIMUM UNDEFLECTED OUTPUT SIGNAL WITH 0.010" DIAMETER DROPPING RESTRICTOR		40% OF SUPPLY PRESSURE
TEMPERATURE LIMITS		0 TO 150F (-17 TO 65C)
MATERIAL	BODY	BRASS
	SPRING	STAINLESS STEEL
	ROD	NYLON

Fig. 2-28: Spring Sensor (AIR LOGIC CO.)

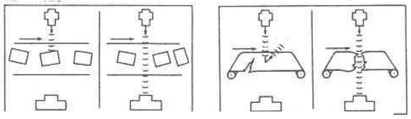


Fig. 13-27 : Ultrasonic sensor, and several applications

CHAPTER 14

FRAFCET METHODE

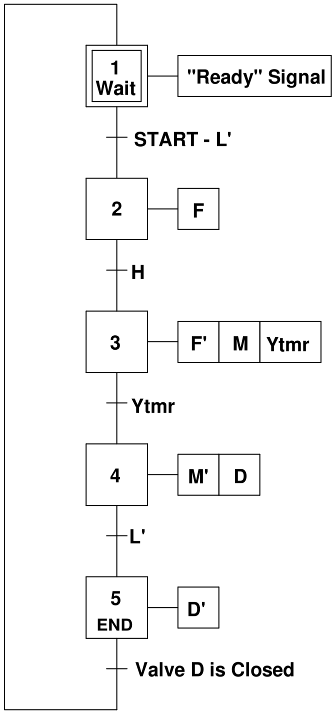


Fig. 14-1 : GRAFCET for Automated Mixing System (Fig. 5-11)

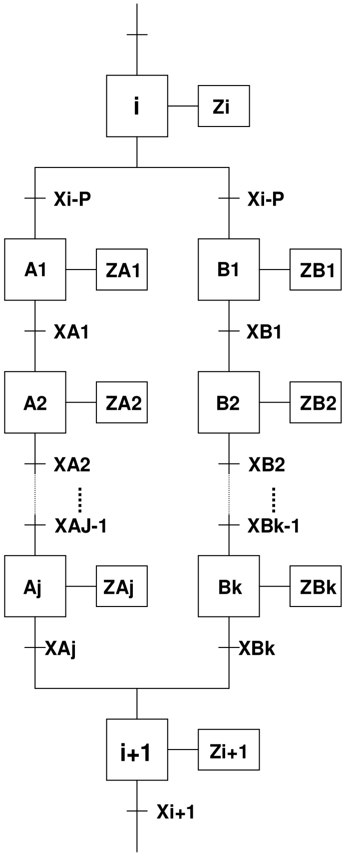


Fig. 14-2 : GRAFCET for Alternate Parallel Paths

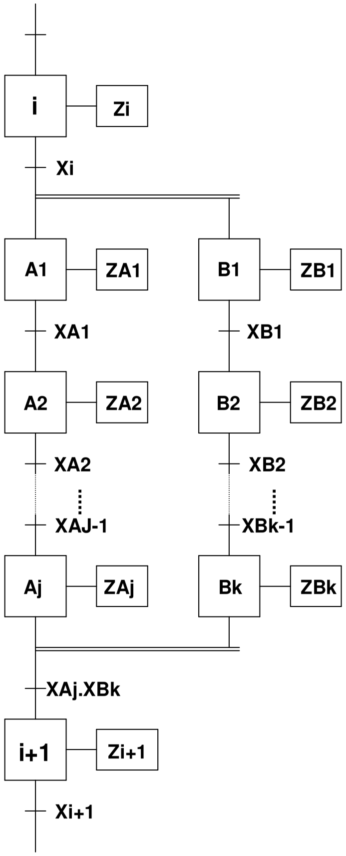
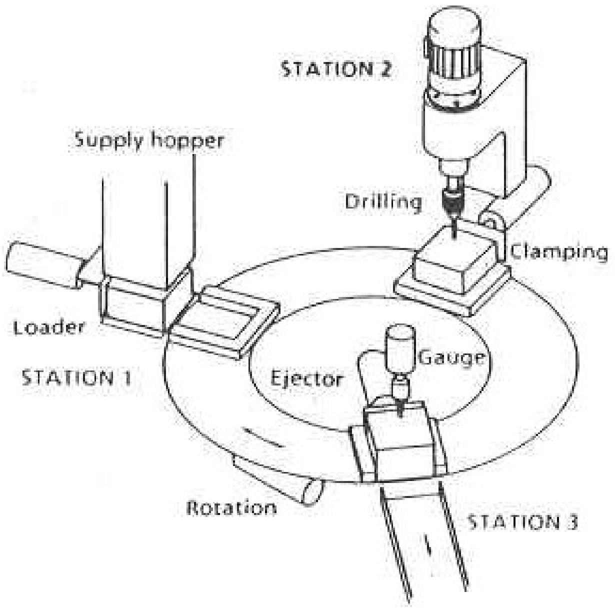


Fig. 14-3 : GRAFCET for Simultaneous Parallel Paths



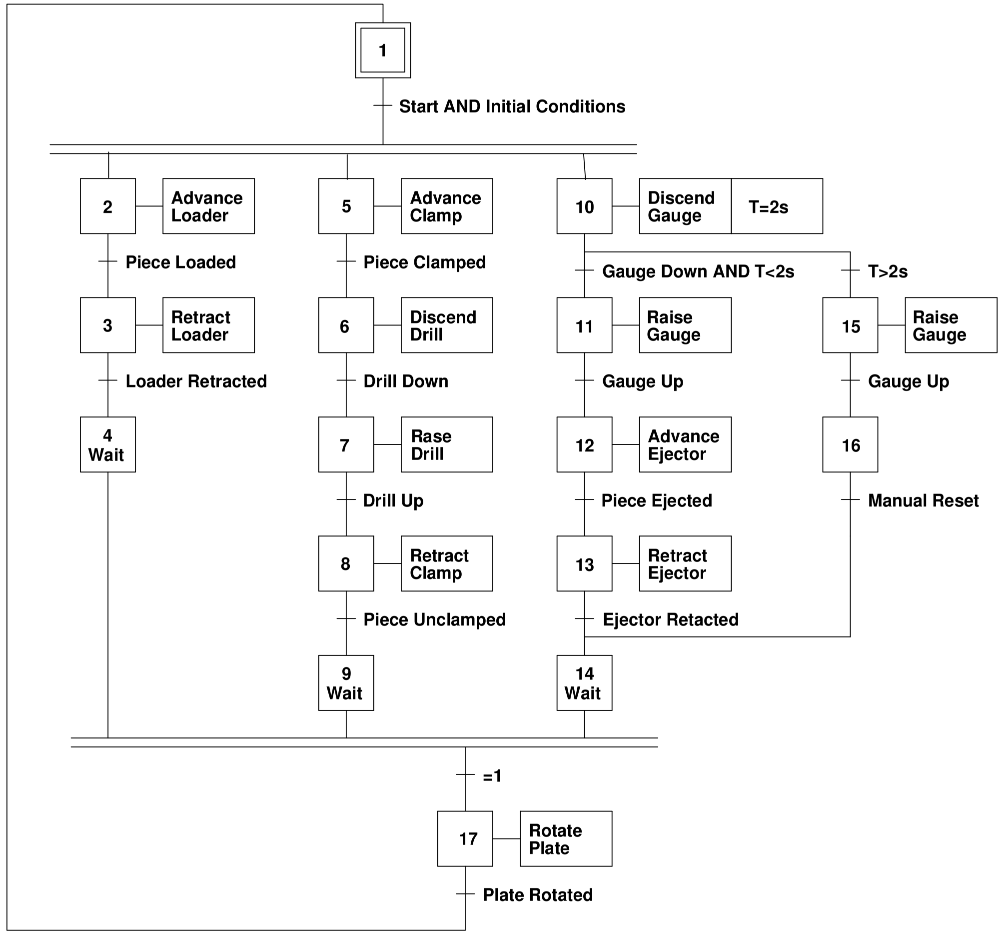


Fig. 14-4 : GRAFCET for Drilling Station

(From Telemechanique TSX T607 Manual)