

# ĐIỀU KHIỂN LOGIC VÀ PLC

TDH-VD-BK

1

1

## Nội dung

1. Cơ sở cho Điều khiển logic
2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp
3. Tổng hợp mạch logic tuần tự
4. Tổng quan về PLC
5. Kỹ thuật lập trình PLC

TDH-VD-BK

2

2

## Nội dung

### **1. Cơ sở cho Điều khiển logic**

2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp
3. Tổng hợp mạch logic tuần tự
4. Tổng quan về PLC
5. Kỹ thuật lập trình PLC

TDH-VD-BK

3

3

## 1. Cơ sở cho điều khiển logic

- 1.1. Khái niệm về điều khiển logic
- 1.2. Đại số logic
- 1.3. Biểu diễn hàm logic

TDH-VD-BK

4

4

# 1. Cơ sở cho điều khiển logic

## 1.1. Khái niệm về điều khiển logic

1.2. Đại số logic

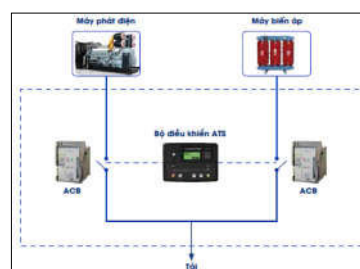
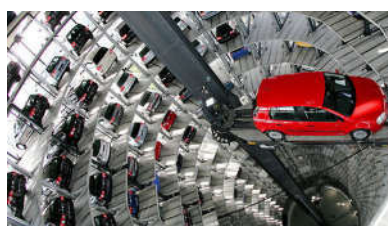
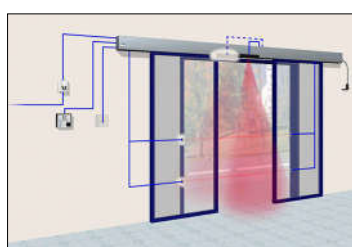
1.3. Biểu diễn hàm logic

TDH-VD-BK

5

5

## 1.1. Khái niệm về Điều khiển logic



TDH-VD-BK

6

6

## 1.1. Khái niệm về Điều khiển logic



TDH-VD-BK

7

7

## 1.1. Khái niệm về Điều khiển logic

- Điều khiển logic giải quyết các vấn đề
  - Hệ thống có các chế độ làm việc khác nhau, tuân theo lệnh điều khiển từ bên ngoài
  - Chuyển từ chế độ này sang chế độ khác theo một trình tự, điều kiện xác định
  - Đảm bảo trình tự thời gian và sự tương tác giữa các bộ phận
  - Phản ứng tức thời trước một số sự kiện

TDH-VD-BK

8

8

## 1.1. Khái niệm về Điều khiển logic

- Các lĩnh vực nghiên cứu điều khiển logic
  - Khoa học máy tính (Computer Science)
  - Lập trình (Programming)
  - Mô phỏng (Simulation)
  - Truyền thông (Communication)
  - **Các hệ thống điều khiển công nghiệp (Industrial Control)**

TDH-VD-BK

9

9

## 1.1. Khái niệm về Điều khiển logic

- Mô hình hóa hệ thống điều khiển logic
  - **Đại số logic (Boolean Algebra)**
  - Automat hữu hạn (Finite State Machine)
  - Statechart
  - **GRAFCET**
  - Petri net

TDH-VD-BK

10

10

# 1. Cơ sở cho điều khiển logic

1.1. Khái niệm về điều khiển logic

**1.2. Đại số logic**

1.3. Biểu diễn hàm logic

TDH-VD-BK

11

11

## 1.2. Đại số logic

- Các sự vật hiện tượng thường được biểu hiện ở hai mặt đối lập:
  - Trong cuộc sống: đúng/sai, có/không, tốt/xấu, sạch/bẩn, đỡ/trượt,
  - Trong kỹ thuật: đóng/cắt, bật/tắt, chạy/dừng
- Để biểu diễn (lượng hóa) trạng thái đối lập: 0 và 1.
- Đại số logic (Đại số Boolean) để nghiên cứu các sự vật, hiện tượng có 2 trạng thái đối lập

TDH-VD-BK

12

12

## 1.2. Đại số logic

- Biến logic:  $x \in [0, 1]$
- Hàm logic :  $f(x_1, x_2, \dots, x_n) \in [0, 1]$

với  $x_1, x_2, \dots, x_n \in [0, 1]$

– Ví dụ: Hàm 1 biến  $f(x)$ :  $f(x) = x$

$$f(x) = \bar{x}$$

$$f(x) = x + \bar{x}$$

$$f(x) = x.\bar{x}$$

Hàm 2 biến  $f(x_1, x_2)$ :  $f(x_1, x_2) = x_1 + x_2$

$$f(x_1, x_2) = \bar{x}_1 x_2 + x_1 \bar{x}_2$$

TDH-VD-BK

13

13

## 1.2. Đại số logic

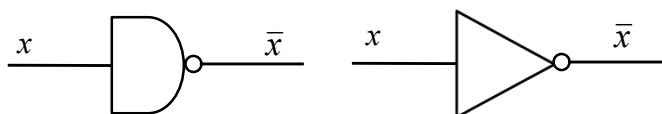
- Các phép toán logic cơ bản

– Phép nghịch đảo: **NOT**

- Bảng giá trị:

$x$	$f(x) = \bar{x}$
1	0
0	1

- Ký hiệu



TDH-VD-BK

14

14

## 1.2. Đại số logic

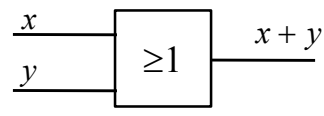
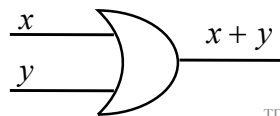
- Các phép toán logic cơ bản

- Phép cộng: **OR**

- Bảng giá trị:

x	y	$f(x,y) = x + y$
0	0	0
0	1	1
1	0	1
1	1	1

- Ký hiệu



TDH-VD-BK

15

15

## 1.2. Đại số logic

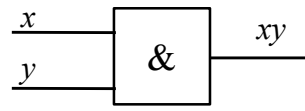
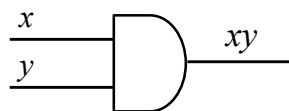
- Các phép toán logic cơ bản

- Phép nhân: **AND**

- Bảng giá trị:

x	y	$f(x,y) = xy$
0	0	0
0	1	0
1	0	0
1	1	1

- Ký hiệu



TDH-VD-BK

16

16



## 1.2. Đại số logic

- Các tính chất của các phép toán logic

– Giao hoán :  $x+y = y+x$

$$xy = yx$$

– Kết hợp:  $x+y+z = (x+y)+z = x+(y+z)$

$$xyz = (xy)z = x(yz)$$

– Phân phối:  $x(y+z) = xy+xz$

$$x+yz = (x+y)(x+z)$$

– Luật De Morgan:

$$\overline{x_1 + x_2 + \dots + x_n} = \overline{x_1} \cdot \overline{x_2} \cdot \dots \cdot \overline{x_n}$$

$$\overline{x_1 \cdot x_2 \cdot \dots \cdot x_n} = \overline{x_1} + \overline{x_2} + \dots + \overline{x_n}$$

17

17

## 1.2. Đại số logic

- Một số hệ thức cơ bản thường gặp

1	$x+0 = x$ $x \cdot 1 = x$
2	$x \cdot 0 = 0$ $x+1 = 1$
3	$x+x = x$ $x \cdot x = x$
4	$x + \bar{x} = 1$ $x \cdot \bar{x} = 0$
5	$x+xy = x$ $x \cdot (x+y) = x$
6	$xy + x\bar{y} = x$ $(x+y)(x+\bar{y}) = x$

Chú ý: Tính đối ngẫu (duality) của các hệ thức logic

18

18

# 1. Cơ sở cho điều khiển logic

1.1. Khái niệm về điều khiển logic

1.2. Đại số logic

**1.3. Biểu diễn hàm logic**

TDH-VD-BK

19

19

## 1.3. Biểu diễn hàm logic

- Bảng chân lý

$x_1$	$x_2$	$x_3$	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	“x”
0	1	1	“x”
1	0	0	0
1	0	1	1
1	1	0	“x”
1	1	1	1

Dấu “x” là giá trị hàm không xác định, có thể nhận giá trị 0 hoặc 1

TDH-VD-BK

20

20

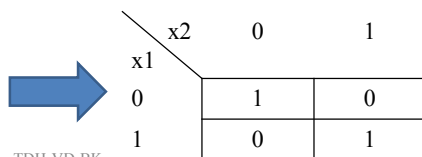
## 1.3. Biểu diễn hàm logic

### • Bảng Các nô (Carnough map)

- Biểu diễn hàm logic  $n$  biến cần thành lập một bảng có  $2^n$  ô, mỗi ô tương ứng với 1 tổ hợp biến.
- Các ô cạnh nhau hoặc đối xứng nhau chỉ cho phép khác nhau về giá trị của 1 biến.
- Trong các ô ghi giá trị của hàm tương ứng với giá trị của tổ hợp biến đó.

Ví dụ:

x1	x2	f(x1,x2)
0	0	1
0	1	0
1	0	0
1	1	1

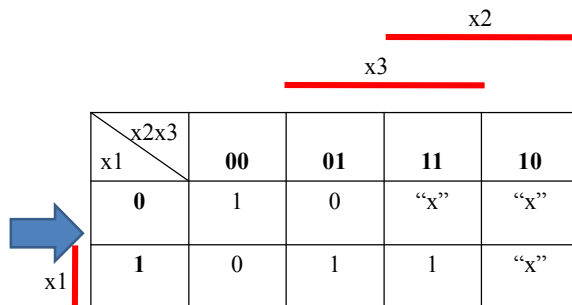


TDH-VD-BK

21

21

x1	x2	x3	f(x1,x2,x3)
0	0	0	1
0	0	1	0
0	1	0	"X"
0	1	1	"X"
1	0	0	0
1	0	1	1
1	1	0	"X"
1	1	1	1



TDH-VD-BK

22

22

x1

x2

		<div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x3</div> <div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x4</div>			
	x1x2 \ x3x4	00	01	11	10
	00				
	01				
	11				
	10				

x1

x2

		<div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x3</div> <div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x4</div> <div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x5</div> <div style="border-bottom: 2px solid red; padding-bottom: 2px; margin-bottom: 2px;">x5</div>							
	x1x2 \ x3x4x5	000	001	011	010	110	111	101	100
	00								
	01								
	11								
	10								

TÜBİTAK

23

23

## 1.3. Biểu diễn hàm logic

- Sơ đồ rơ le – tiếp điểm

Thiết bị	Loại	Ký hiệu
Nút ấn	Thường mở	
	Thường đóng	
Công tắc hành trình	Thường mở	
	Thường đóng	
Rơ le	Cuộn dây	
	Tiếp điểm thường mở	
	Tiếp điểm thường đóng	

24

## Biểu diễn hàm logic

- Sơ đồ rơ le - tiếp điểm
  - Hai dây thể hiện nguồn cấp
  - Lựa chọn ký hiệu biến tương ứng với thiết bị vật lý (nút ấn, công tắc hành trình hay tiếp điểm rơ le)
  - Biến ở trạng thái thường: tiếp điểm thường mở
  - Biến ở trạng thái đảo: tiếp điểm thường đóng
  - Cộng logic: đấu song song
  - Nhân logic: đấu nối tiếp
  - Đầu ra: cuộn dây rơ le đấu nối tiếp với tổ hợp biểu diễn các biến đầu vào

TDH-VD-BK

25

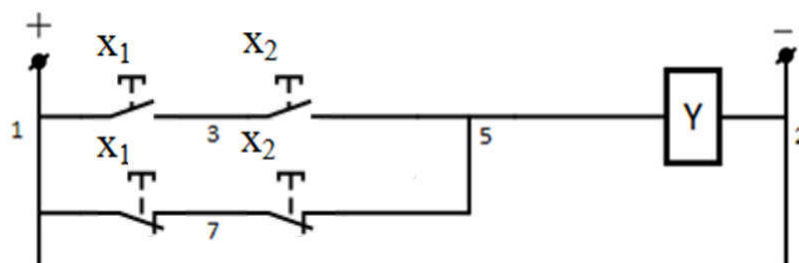
25

### 1.3. Biểu diễn hàm logic

- Sơ đồ rơ le – tiếp điểm

Ví dụ:

$$Y = f(x_1, x_2) = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$



TDH-VD-BK

26

26

# ĐIỀU KHIỂN LOGIC VÀ PLC

TDH-VD-BK

1

1

## Nội dung

1. Cơ sở cho Điều khiển logic
- 2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp**
3. Tổng hợp mạch logic tuần tự
4. Tổng quan về PLC
5. Kỹ thuật lập trình PLC

TDH-VD-BK

2

2

## 2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp

### 2.1. Khái niệm mạch logic tổ hợp

### 2.2. Tổng hợp mạch logic tổ hợp

- Dạng tổng chuẩn đầy đủ
- Dạng tích chuẩn đầy đủ

### 2.3. Tối thiểu hóa mạch logic tổ hợp

- Phương pháp đại số
- Phương pháp bảng Các nô (Carnough map)
- Phương pháp Quine Mc. Clusky

TDH-VD-BK

3

3

## 2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp

### 2.1. Khái niệm mạch logic tổ hợp

### 2.2. Tổng hợp mạch logic tổ hợp

- Dạng tổng chuẩn đầy đủ
- Dạng tích chuẩn đầy đủ

### 2.3. Tối thiểu hóa mạch logic tổ hợp

- Phương pháp đại số
- Phương pháp bảng Các nô (Carnough map )
- Phương pháp Quine Mc. Clusky

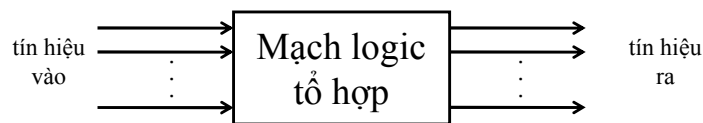
TDH-VD-BK

4

4

## 2.1. Khái niệm về mạch logic tổ hợp

- Định nghĩa: Mạch logic tổ hợp là mạch logic mà tín hiệu ra của mạch chỉ phụ thuộc vào tín hiệu đầu vào, không phụ thuộc vào thứ tự, thời gian tác động của tín hiệu vào
- Tính chất
  - Không có nhớ
  - Không có yếu tố thời gian
  - Cùng một tổ hợp tín hiệu vào, tín hiệu ra là duy nhất
  - Mạch vòng hở



TDH-VD-BK

5

5

## 2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp

### 2.1. Khái niệm mạch logic tổ hợp

### 2.2. Tổng hợp mạch logic tổ hợp

- Dạng tổng chuẩn đầy đủ
- Dạng tích chuẩn đầy đủ

### 2.3. Tối thiểu hóa mạch logic tổ hợp

- Phương pháp đại số
- Phương pháp bảng Các nô (Carnough map )
- Phương pháp Quine Mc. Clusky

TDH-VD-BK

6

6



## 2.2. Tổng hợp mạch logic tổ hợp

### – Dạng tổng chuẩn đầy đủ

- Chỉ quan tâm đến tổ hợp các giá trị của biến làm cho hàm có giá trị 1. Mỗi tổ hợp này tương ứng với một tích của tất cả các biến.
- Trong mỗi tích, các biến có giá trị 1 thì được biểu diễn ở trạng thái thường, các biến có giá trị 0 thì được biểu diễn ở trạng thái phủ định.
- Hàm logic dạng tổng chuẩn đầy đủ sẽ là tổng các tích đó

x	y	f(x,y)
0	0	1
0	1	0
1	0	0
1	1	1

$$f(x, y) = \bar{x} \bar{y} + xy$$

TDH-VD-BK

7

7

### – Chú ý: Cách ký hiệu rút gọn của hàm logic

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$$

$$f(x_1, x_2, x_3) = \sum (0, 2, 5, 6, 7) + N(4, 3)$$

Thập phân	x1	x2	x3	f(x1,x2,x3)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

if '4' == x  
&& '3' == x

TDH-VD-BK

8

8

## 2.2. Tổng hợp mạch logic tổ hợp

### – Dạng tích chuẩn đầy đủ

- Chỉ quan tâm đến tổ hợp các giá trị của biến làm cho hàm có giá trị 0. Mỗi tổ hợp này tương ứng với một tổng của tất cả các biến.
- Trong mỗi tổng, các biến có giá trị 0 thì được biểu diễn ở trạng thái thường, các biến có giá trị 1 thì được biểu diễn ở trạng thái phủ định.
- Hàm logic dạng tích chuẩn đầy đủ sẽ là tích các tổng đó

x	y	f(x,y)
0	0	1
0	1	0
1	0	0
1	1	1

$$f(x, y) = (x + \bar{y})(\bar{x} + y)$$

$$\pi(2,3)$$

TDH-VD-BK

9

9

## 2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp

### 2.1. Khái niệm mạch logic tổ hợp

### 2.2. Tổng hợp mạch logic tổ hợp

- Dạng tổng chuẩn đầy đủ
- Dạng tích chuẩn đầy đủ

### 2.3. Tối thiểu hóa mạch logic tổ hợp

- Phương pháp đại số
- Phương pháp bảng Các nô (Carnough map )
- Phương pháp Quine Mc. Clusky

TDH-VD-BK

10

10

## 2.3. Tối thiểu hóa hàm logic

- Phương pháp biến đổi đại số

- Dựa vào các hệ thức cơ bản

$$\begin{aligned}
 f(a, b) &= \bar{a}b + ab + a\bar{b} \\
 &= (\bar{a}b + ab) + (ab + a\bar{b}) \\
 &= (\bar{a} + a)b + a(b + \bar{b}) \\
 &= b + a
 \end{aligned}$$

- Nhược điểm: không biết rõ đã tối thiểu chưa

TDH-VD-BK

11

11

## 2.3. Tối thiểu hóa hàm logic

- Phương pháp bảng Các ô

- Biểu diễn hàm đã cho dưới dạng bảng Các ô

- Nhóm các ô có giá trị 1 và không xác định ở cạnh nhau hoặc đối xứng nhau thành các vòng:

- Số ô trong 1 vòng là  $2^m$ , m lớn nhất có thể
    - Các vòng có thể giao nhau nhưng không được trùm lên nhau.
    - Các vòng phải phủ hết các ô có giá trị 1
    - Số vòng phải là tối thiểu.

- Mỗi vòng tương ứng với tích các biến có giá trị không thay đổi trong vòng đó với biểu diễn tương ứng với giá trị của các biến.

- Hàm rút gọn bằng tổng các tích tương ứng với các vòng.

12

12

– Ví dụ 1:

$$f(x_1, x_2, x_3) = \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + \overline{x_1} x_2 x_3 + \overline{x_1} x_2 x_3$$

x1	x2	x3	f(x1,x2,x3)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

x1 \ x2x3	00	01	11	10
0	1	0	0	1
1	0	1	1	1

$$f(x_1, x_2, x_3) = \overline{x_1} \overline{x_3} + x_1 x_3 + x_1 x_2$$

x1 \ x2x3	00	01	11	10
0	1	0	0	1
1	0	1	1	1

$$f(x_1, x_2, x_3) = \overline{x_1} \overline{x_3} + x_1 x_3 + x_2 \overline{x_3}$$

13

13

– Ví dụ 2:

$$f(x_1, x_2, x_3, x_4) = \sum \left( \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + x_1 \overline{x_2} \overline{x_3} \overline{x_4} + x_1 \overline{x_2} \overline{x_3} x_4 + x_1 \overline{x_2} x_3 \overline{x_4} + x_1 \overline{x_2} x_3 x_4 + x_1 x_2 \overline{x_3} \overline{x_4} + x_1 x_2 \overline{x_3} x_4 \right)$$

		x3			
		x4			
x1	x2 \ x3x4	00	01	11	10
	00	1	0	0	1
	01	0	0	0	0
	11	0	1	1	0
	10	1	1	1	1

$$f(x_1, x_2, x_3, x_4) = \overline{x_2} \overline{x_4} + x_1 x_4 + x_1 \overline{x_2}$$

CHƯA  
TỐI  
THIỆU

TDH-VD-BK

14

14

$x_1x_2 \backslash x_3x_4$	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	1	1	0
10	1	1	1	1

$$f(x_1, x_2, x_3, x_4) = \bar{x}_2 \bar{x}_4 + x_1 x_4$$

TDH-VD-BK

15

15

## 2.3. Tối thiểu hóa hàm logic

### • Phương pháp Quine Mc. Clusky

- Ghi các tổ hợp biến theo mã nhị phân (đảo = 0)
- Nhóm các tổ hợp biến theo số chữ số 1 trong biểu diễn nhị phân, nhóm  $i$  có  $i$  chữ số 1
- Ghép tổ hợp nhóm thứ  $i$  với nhóm  $i+1$  nếu chúng chỉ khác nhau 1 bit ở cùng 1 vị trí. Đánh dấu “-” vào vị trí bit đổi trị trong tổ hợp mới hình thành. Đánh dấu “\*” vào các tổ hợp đã tham gia ghép, dấu “√” vào các tổ hợp không thể ghép
- Lặp lại 2 bước trên đến khi không kết hợp được
- Lập bảng phủ tối thiểu: chọn số tổ hợp không thể ghép tối thiểu để phủ hết số tổ hợp ban đầu
- Hàm tối thiểu bằng tổng các tích ứng với các tổ hợp không thể ghép được lựa chọn trong bảng phủ tối thiểu

TDH-VD-BK

16

16



Bảng phủ

	0000	0100	0101	0110	0111	1000	1001	1010	1101	1111
0-00	x	x								
-000	x					x				
100-						x	x			
10-0						x		x		
1-01							x		x	
01--		x	x	x	x					
-1-1			x		x					x

$$f(a,b,c,d) = \bar{a}\bar{c}\bar{d} + a\bar{b}\bar{c} + a\bar{b}\bar{d} + \bar{a}b + bd$$

TDH-VD-BK

19

19

- Bài tập về nhà:
- Rút gọn dùng bảng Các nô:

$$f(x,y,z) = \sum (0,1,6,7)$$

$$f(w,x,y,z) = \sum (1,3,7,9,11,15)$$

$$f(v,w,x,y,z) = \sum (0,4,18,19,23,27,28,29,31)$$

- Rút gọn dùng phương pháp Quine Mc.Clusky

$$f(x,y,z) = \sum (2,3,4,5)$$

$$f(w,x,y,z) = \sum (0,1,4,5,12,13)$$

$$f(w,x,y,z) = \sum (1,4,5,7,8,9,13,14,15)$$

TDH-VD-BK

20

20

# ĐIỀU KHIỂN LOGIC VÀ PLC

TDH-VD-BK

1

1

## Nội dung

1. Cơ sở cho Điều khiển logic
2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp
- 3. Tổng hợp mạch logic tuần tự**
4. Tổng quan về PLC
5. Kỹ thuật lập trình PLC

TDH-VD-BK

2

2



### 3. Tổng hợp mạch logic tuần tự

#### 3.1. Khái niệm mạch logic tuần tự

- Định nghĩa
- Tính chất
- Phân loại
- Biểu diễn bằng đồ thị thời gian

#### 3.2. Tổng hợp mạch logic tuần tự

- Phương pháp ma trận trạng thái
- Phương pháp GRAFCET

TDH-VD-BK

3

3

### 3. Tổng hợp mạch logic tuần tự

#### 3.1. Khái niệm mạch logic tuần tự

- Định nghĩa
- Tính chất
- Phân loại
- Biểu diễn bằng đồ thị thời gian

#### 3.2. Tổng hợp mạch logic tuần tự

- Phương pháp ma trận trạng thái
- Phương pháp GRAFCET

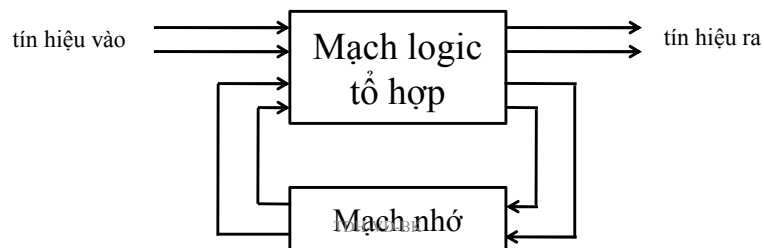
TDH-VD-BK

4

4

### 3.1. Khái niệm về mạch logic tuần tự

- Định nghĩa: Mạch logic tuần tự là mạch logic mà tín hiệu ra của mạch không những phụ thuộc vào tín hiệu đầu vào, mà còn phụ thuộc vào thứ tự, thời gian tác động của tín hiệu vào
- Tính chất
  - Có nhớ
  - Có yếu tố thời gian
  - Cùng 1 tín hiệu vào, tín hiệu ra có thể khác nhau (các trạng thái trong hay trạng thái làm việc)
  - Mạch vòng kín (có phản hồi)



5

5

### 3.1. Khái niệm về mạch logic tuần tự

- Phân loại
  - Mạch logic tuần tự đồng bộ: việc chuyển trạng thái trong mạch không những chỉ phụ thuộc vào tín hiệu đầu vào, trạng thái trước đó, mà còn phụ thuộc vào xung đồng bộ
    - Dùng phổ biến trong máy tính (môn ĐT số)
  - Mạch logic tuần tự không đồng bộ: việc chuyển trạng thái trong mạch chỉ phụ thuộc vào tín hiệu đầu vào, trạng thái trước đó
    - Không có tín hiệu đồng bộ
    - Thường gặp trong công nghệ của các máy sản xuất công nghiệp

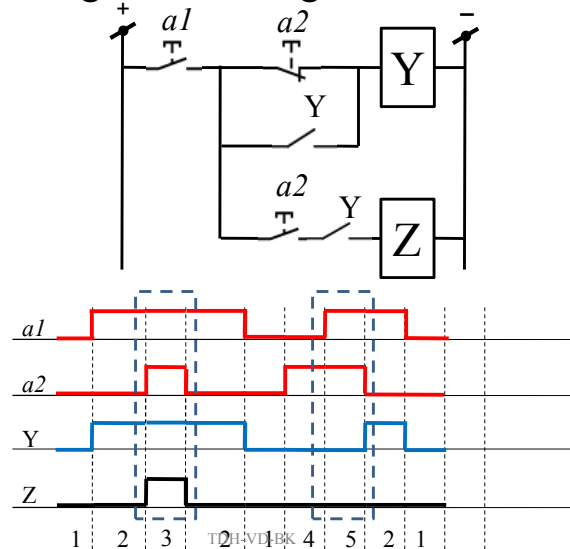
TDH-VD-BK

6

6

### 3.1. Khái niệm về mạch logic tuần tự

- Biểu diễn bằng đồ thị thời gian



7

## 3. Tổng hợp mạch logic tuần tự

### 3.1. Khái niệm mạch logic tuần tự

- Định nghĩa
- Tính chất
- Phân loại
- Biểu diễn bằng đồ thị thời gian

### 3.2. Tổng hợp mạch logic tuần tự

- Phương pháp ma trận trạng thái
- Phương pháp GRAFCET

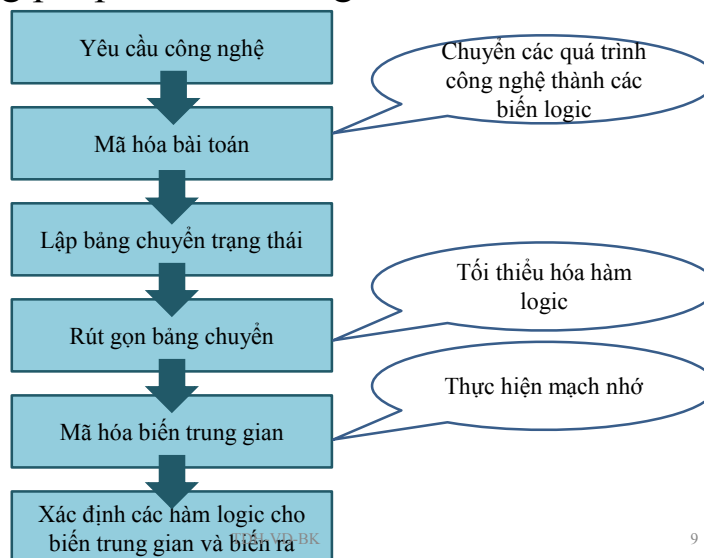
TDH-VD-BK

8

8

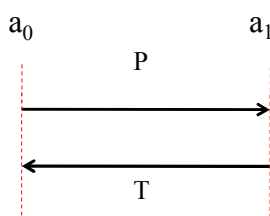
### 3.2. Tổng hợp mạch logic tuần tự

- Phương pháp ma trận trạng thái



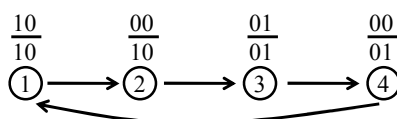
9

#### Ví dụ 1:



- Mã hóa bài toán:

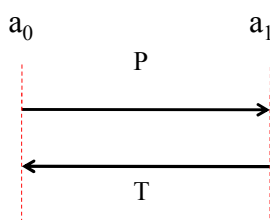
- Xác định các biến vào ra:  $\frac{\text{Vào}}{\text{Ra}} = \frac{a_0 a_1}{PT}$
- Graph chuyển trạng thái:



TDH-VD-BK

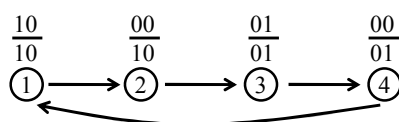
10

10

Ví dụ 1:

## ▪ Mã hóa bài toán:

- Xác định các biến vào ra:  $\frac{\text{Vào}}{\text{Ra}} = \frac{a_0 a_1}{PT}$
- Graph chuyển trạng thái:

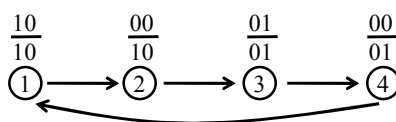


TDH-VD-BK

11

11

## ▪ Lập bảng chuyển trạng thái MI:



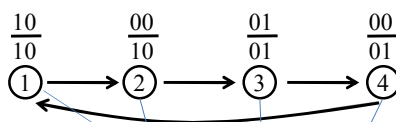
Trạng thái	Tín hiệu vào: $a_0 a_1$				Tín hiệu ra	
	00	01	11	10	P	T
① (sang phải)					1	0
② (trên đường sang phải)					1	0
③ (sang trái)					0	1
④ (trên đường sang trái)					0	1

TDH-VD-BK

12

12

▪ Lập bảng chuyển trạng thái MI: các đỉnh



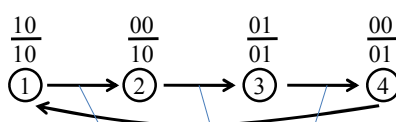
Trạng thái	Tín hiệu vào: $a_0a_1$				Tín hiệu ra	
	00	01	11	10	P	T
① (sang phải)				①	1	0
② (trên đường sang phải)	②				1	0
③ (sang trái)		③			0	1
④ (trên đường sang trái)	④				0	1

TDH-VD-BK

13

13

▪ Lập bảng chuyển trạng thái MI: các cung có hướng



Trạng thái	Tín hiệu vào: $a_0a_1$				Tín hiệu ra	
	00	01	11	10	P	T
① (sang phải)	2			①	1	0
② (trên đường sang phải)	②	3			1	0
③ (sang trái)	4	③			0	1
④ (trên đường sang trái)	④			1	0	1

TDH-VD-BK

14

14

### ▪ Rút gọn bảng chuyển

(Lập bảng chuyển trạng thái M II: nhập hàng của M I)

#### ➤ Quy tắc nhập hàng:

- ✓ Trên cùng 1 cột biến vào, các hàng phải có cùng số ký hiệu trạng thái hoặc là giá trị trống.
- ✓ Không quan tâm đến giá trị biến đầu ra, nhưng ưu tiên nhập các hàng có đầu ra giống nhau.
- ✓ Số hàng nhập nhiều nhất có thể
- ✓ Trạng thái ổn định nhập với không ổn định sẽ ghi trạng thái ổn định.
- ✓ Trạng thái (/không) ổn định nhập với 1 ô trống sẽ ghi trạng thái (/không) ổn định

TDH-VD-BK

15

15

Bảng M I

Trạng thái	Tín hiệu vào: $a_0a_1$				Tín hiệu ra	
		$a_1$		$a_0$	P	T
	00	01	11	10		
① (sang phải)	2			①	1	0
② (trên đường sang phải)	②	3			1	0
③ (sang trái)	4	③			0	1
④ (trên đường sang trái)	④			1	0	1



Bảng M II

	$a_1$		$a_0$	
① + ②	② <sup>10</sup>	3		① <sup>10</sup>
③ + ④	④ <sup>01</sup>	③ <sup>01</sup>		1

TDH-VD-BK

16

16

▪ Xác định và mã hóa biến trung gian

– Số lượng biến trung gian tối thiểu  $S_{\min}$

$$2^{S_{\min}} \geq N \quad (N: \text{số hàng của M II})$$

–  $N = 2 \Rightarrow S_{\min} = 1 \Rightarrow$  chọn 1 biến trung gian  $X$

– Mã hóa

		$a_1$	$a_0$	
$\textcircled{1} + \textcircled{2}$	$\textcircled{2}$	3	$\textcircled{1}$	$X = 0$
$\textcircled{3} + \textcircled{4}$	$\textcircled{4}$	$\textcircled{3}$	1	$X = 1$

TDH-VD-BK

17

17

▪ Xác định hàm điều khiển cho biến trung gian  $X$ :

Bảng M II

		$a_1$	$a_0$	
$\textcircled{1} + \textcircled{2}$	$\textcircled{2}$	3	$\textcircled{1}$	$X = 0$
$\textcircled{3} + \textcircled{4}$	$\textcircled{4}$	$\textcircled{3}$	1	$X = 1$

Bảng Các nô  
cho biến  $X$

		$a_1$	$a_0$	
	00	01	11	10
	0	1		0
$X$	1	1		0

$$X = a_1 + \bar{a}_0 \cdot X$$

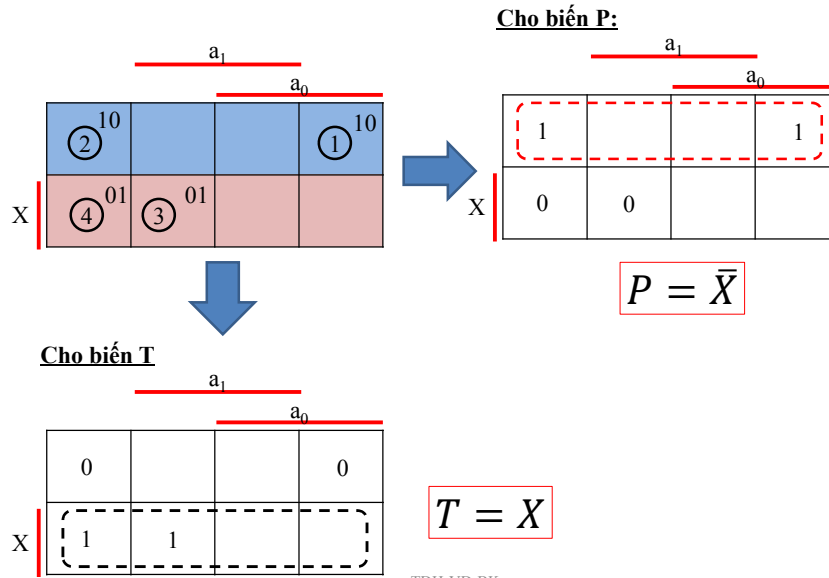
TDH-VD-BK

18

18

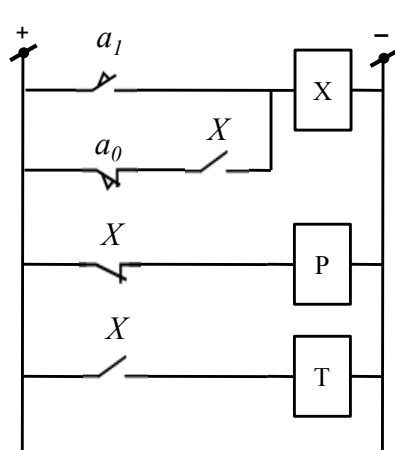


▪ Xác định hàm logic điều khiển các biến ra



19

• Sơ đồ nguyên lý



$$X = a_1 + \bar{a}_0 \cdot X$$

$$P = \bar{X}$$

$$T = X$$

Nếu thay X bằng T, chuyện gì xảy ra?

Trong các hàng của M II, các trạng thái ổn định đều có cùng giá trị đầu ra, có thể cho phép dùng biến ra làm biến trung gian

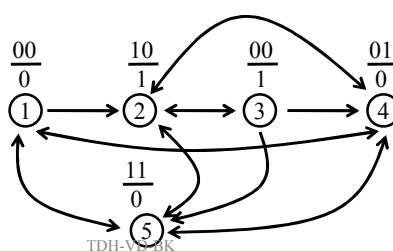
20

20

• Ví dụ 2: 2 nút ấn m và d, 1 thiết bị điện T

- Ấn nút m: đóng điện cho T
- Ấn nút d: cắt điện của T
- 2 nút ấn đồng thời: T ngắt điện

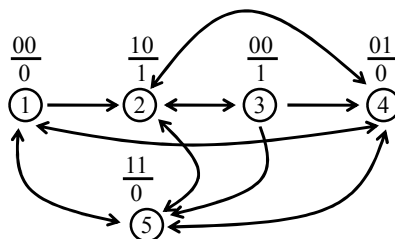
- Chọn các biến vào ra:  $\frac{\text{Vào}}{\text{Ra}} = \frac{md}{T}$
- Graph chuyển trạng thái



TDH-VD-BK

21

21



Bảng M I

Trạng thái	Tín hiệu vào: md				Tín hiệu ra T
	00	01	11	10	
①	①	4	5	2	0
②	3	4	5	②	1
③	③	4	5	2	1
④	1	④	5	2	0
⑤	1	4	⑤	2	0

TDH-VD-BK

22

22

• Bảng chuyển trạng thái M I & M II

Bảng M I

Trạng thái	Tín hiệu vào: md				Tín hiệu ra T
	00	01	11	10	
①	①	4	5	2	0
②	3	4	5	②	1
③	③	4	5	2	1
④	1	④	5	2	0
⑤	1	4	⑤	2	0

Bảng M II

	d			
	m			
① + ④ + ⑤	① <sup>0</sup>	④ <sup>0</sup>	⑤ <sup>0</sup>	2 <sup>1</sup>
② + ③	③ <sup>1</sup>	4 <sup>0</sup>	5 <sup>0</sup>	② <sup>1</sup>

TDH-VD-BK

23

23

• Xác định và mã hóa biến trung gian:

– Smin = 1, chọn biến trung gian là biến ra X = T

	T			
	①	④	⑤	② ③
	d			
	m			
	① <sup>0</sup>	④ <sup>0</sup>	⑤ <sup>0</sup>	2 <sup>1</sup>
T	③ <sup>1</sup>	4 <sup>0</sup>	5 <sup>0</sup>	② <sup>1</sup>
	d			
	m			
	0	0	0	1
T	1	0	0	1

TDH-VD-BK

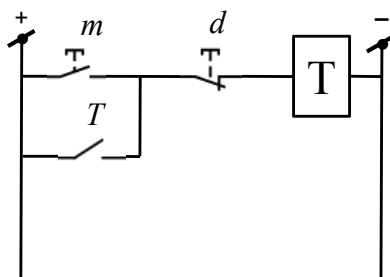
24

$$T = m\bar{d} + T\bar{d}$$

$$= (m + T)\bar{d}$$

24

- Sơ đồ rơ le-tiếp điểm

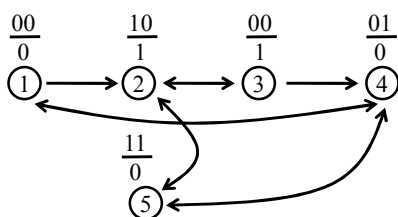


TDH-VD-BK

25

25

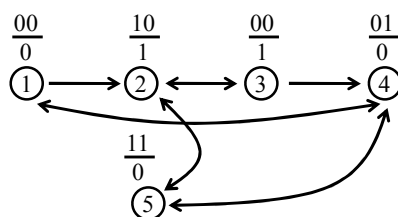
- Ví dụ 2.1: 2 nút ấn m và d, 1 thiết bị điện T



TDH-VD-BK

26

26



Bảng M I

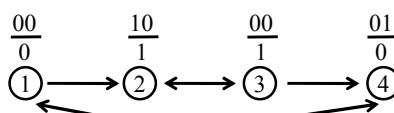
Trạng thái	Tín hiệu vào: md				Tín hiệu ra T
	00	01	11	10	
①	①	4	5	2	0
②	3		5	②	1
③	③	4	5	2	1
④	1	④	5		0
⑤		4	⑤	2	0

TDH-VD-BK

27

27

- Ví dụ 2.2: 2 nút ấn m và d, 1 thiết bị điện T
  - Ấn nút m: đóng điện cho T
  - Ấn nút d: cắt điện của T
- Chọn các biến vào ra:
- Graph chuyển trạng thái  $\frac{\text{Vào}}{\text{Ra}} = \frac{md}{T}$



TDH-VD-BK

28

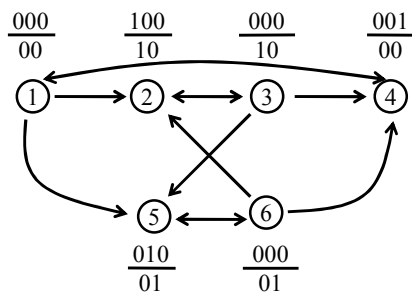
28

- Ví dụ 3: 3 nút ấn a, b và c, động cơ M
  - Ấn nút a: động cơ quay thuận
  - Ấn nút b: động cơ quay ngược
  - Ấn nút c: động cơ dừng
  - Đang quay thuận, ấn b: động cơ quay ngược
  - Đang quay ngược, ấn a: động cơ quay thuận
- Chọn các biến vào ra:  $\frac{\text{Vào}}{\text{Ra}} = \frac{abc}{\text{TN}}$ 
  - Quay thuận: TN = 10
  - Quay ngược: TN = 01
  - Dừng: TN = 00

TDH-VD-BK

29

29



Bảng M I

Trạng thái	Tín hiệu vào: abc								Tín hiệu ra	
	000	001	011	010	110	111	101	100	T	N
①	①	4		5				2	0	0
②	3							②	1	0
③	③	4		5				2	1	0
④	1	④							0	0
⑤	6			⑤					0	1
⑥	⑥	4		5				2	0	1

TDH-VD-BK

30

30

Bảng M II

① + ④	① <sup>00</sup>	④ <sup>00</sup>		5 <sup>01</sup>				2 <sup>10</sup>
② + ③	③ <sup>10</sup>	4 <sup>00</sup>		5 <sup>01</sup>				② <sup>10</sup>
⑤ + ⑥	⑥ <sup>01</sup>	4 <sup>00</sup>		⑤ <sup>01</sup>				2 <sup>10</sup>

Xác định và mã hóa biến trung gian: X ; Y

**Nhận xét:** Biến trung gian trùng với biến đầu ra  $T = X$ ;  $N = Y$

①	④	XY = TN = 00
②	③	XY = TN = 10
⑤	⑥	XY = NT = 01

TDH-VD-BK

31

31

**Chú ý:** Chuyển từ bảng MII sang bảng Các nô

① + ④	① <sup>00</sup>	④ <sup>00</sup>		5 <sup>01</sup>				2 <sup>10</sup>	TN = 00
② + ③	③ <sup>10</sup>	4 <sup>00</sup>		5 <sup>01</sup>				② <sup>10</sup>	TN = 10
⑤ + ⑥	⑥ <sup>01</sup>	4 <sup>00</sup>		⑤ <sup>01</sup>				2 <sup>10</sup>	TN = 01

abc	000	001	011	010	110	111	101	100
TN	00	00		01				10
01	01	00		01				10
11								
10	10	00		01				10

TDH-VD-BK

32

32

abc TN	000	001	011	010	110	111	101	100
00	0	0		0				1
01	0	0		0				1
11								
10	1	0		0				1

$$T = a + T \cdot \bar{b} \cdot \bar{c}$$

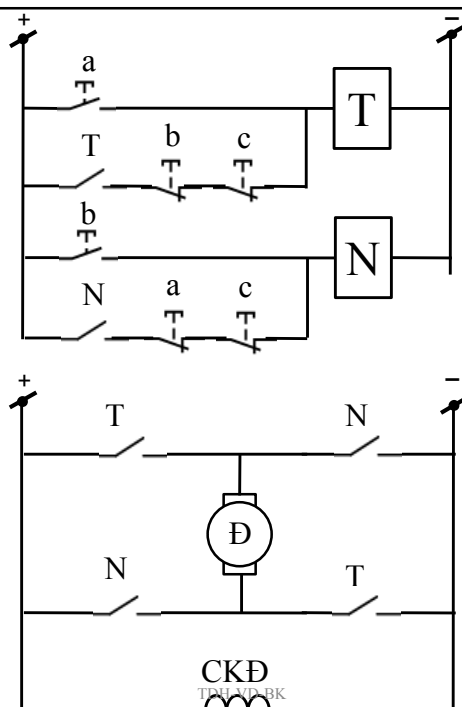
abc TN	000	001	011	010	110	111	101	100
00	0	0		1				0
01	1	0		1				0
11								
10	0	0		1				0

$$N = b + N \cdot \bar{a} \cdot \bar{c}$$

TDH-VD-BK

33

33



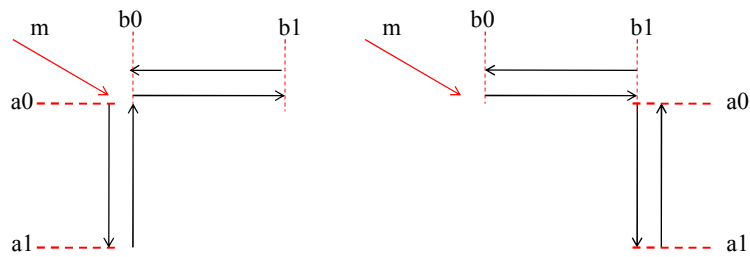
TDH-VD-BK

34

34



- BTVN:
- Cho 3 nút ấn A, B, C điều khiển động cơ M1, M2
  - A: M1 làm việc
  - B: M2 làm việc
  - C: M1, M2 dừng
  - M1 làm việc trước rồi M2 mới làm việc

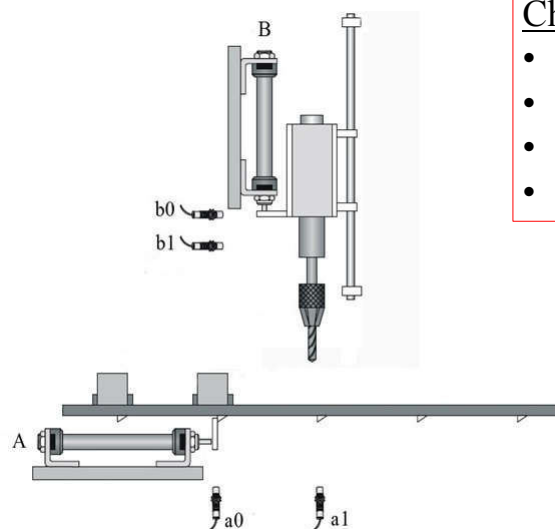


TDH-VD-BK

35

35

- Ví dụ 4:



Chu trình làm việc:

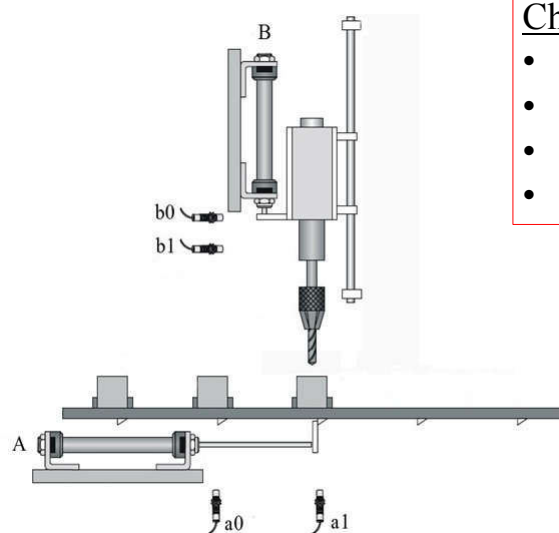
- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

36

36

• Ví dụ 4:



Chu trình làm việc:

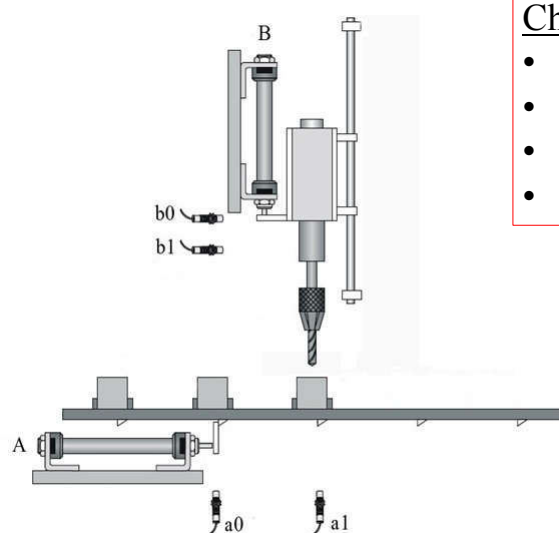
- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

37

37

• Ví dụ 4:



Chu trình làm việc:

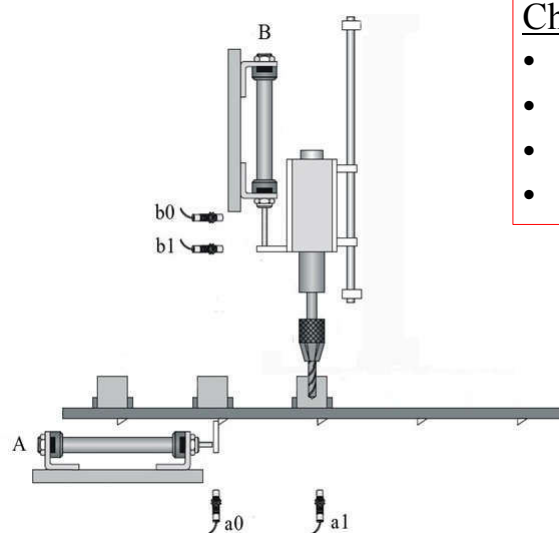
- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

38

38

• Ví dụ 4:



Chu trình làm việc:

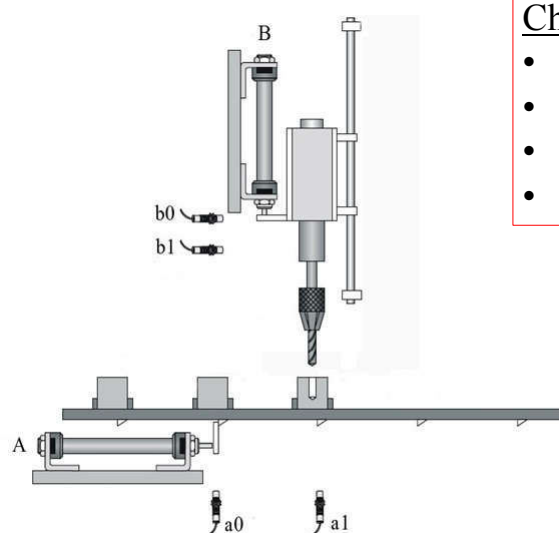
- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

39

39

• Ví dụ 4:



Chu trình làm việc:

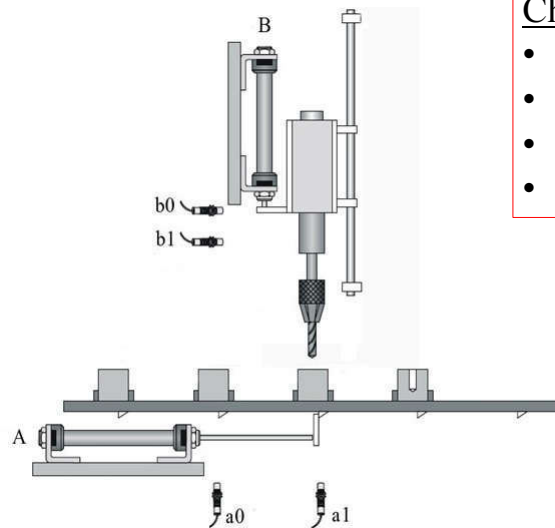
- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

40

40

• Ví dụ 4:



Chu trình làm việc:

- A sang phải (A+)
- A sang trái (A-)
- B đi xuống (B+)
- B đi lên (B-)

TDH-VD-BK

41

41

• Nhận xét:

- Số biến vào lớn
- Có thể rút gọn số biến vào:
  - Chọn a sao cho a1 là tín hiệu đóng (set) của a, a0 là tín hiệu cắt (reset) của a

$$a = a_1 + a \cdot \bar{a}_0$$

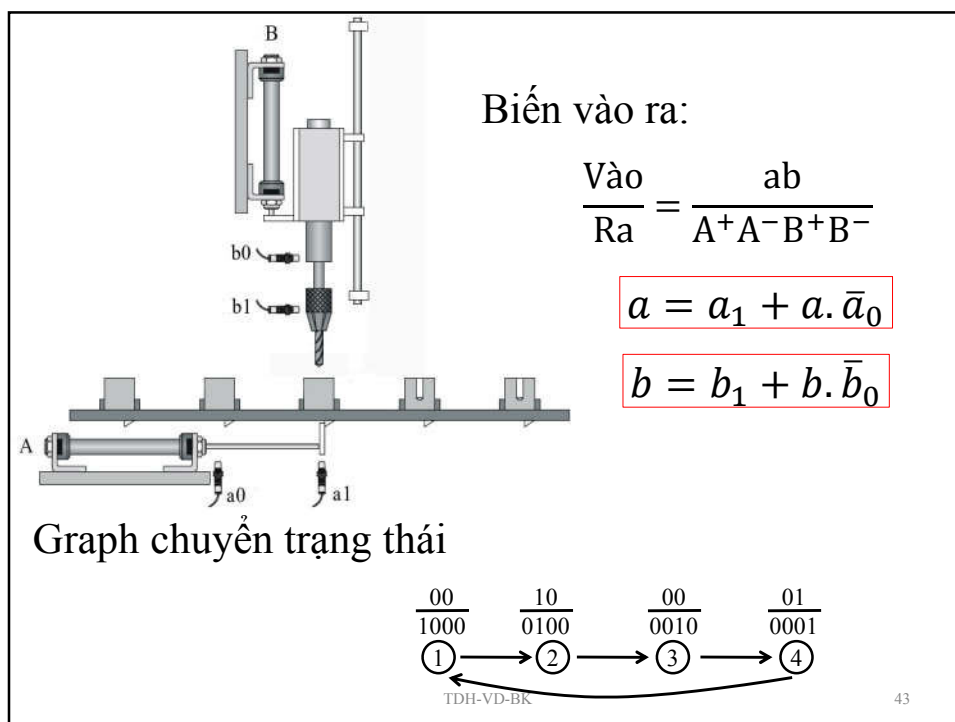
- Chọn b sao cho b1 là tín hiệu đóng (set) của b, b0 là tín hiệu cắt (reset) của b

$$b = b_1 + b \cdot \bar{b}_0$$

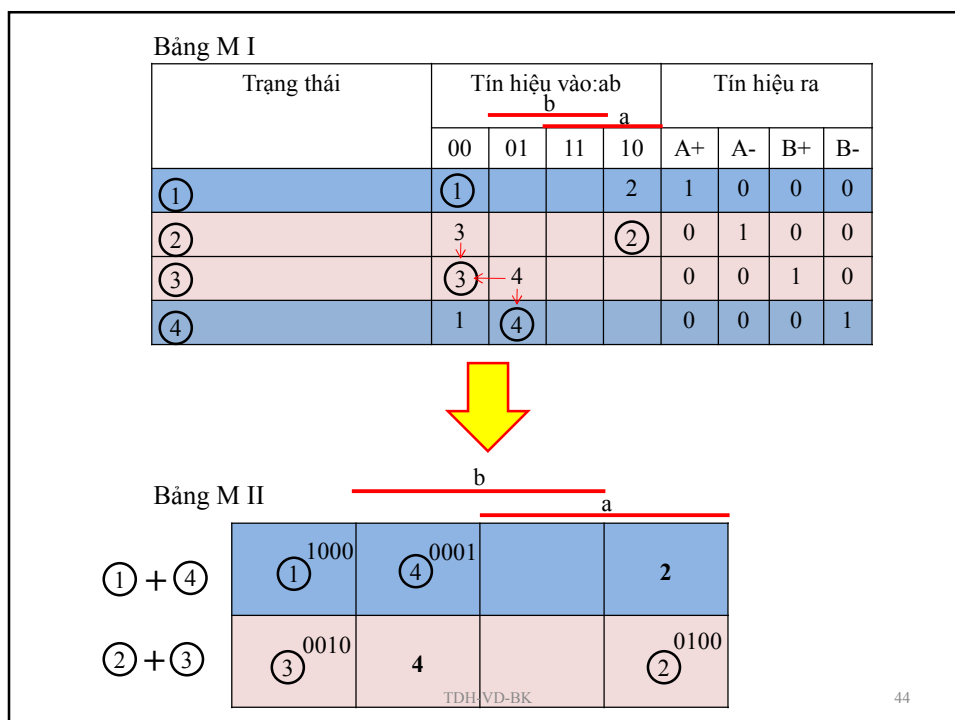
TDH-VD-BK

42

42

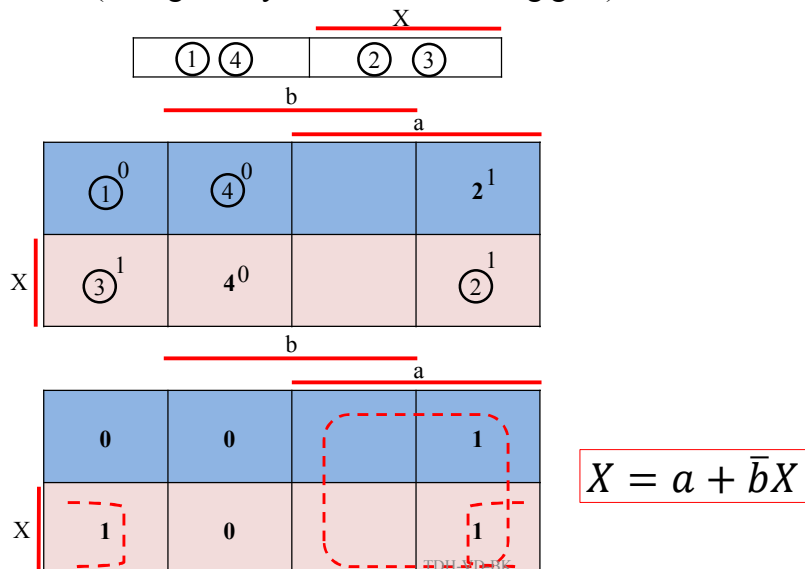


43



44

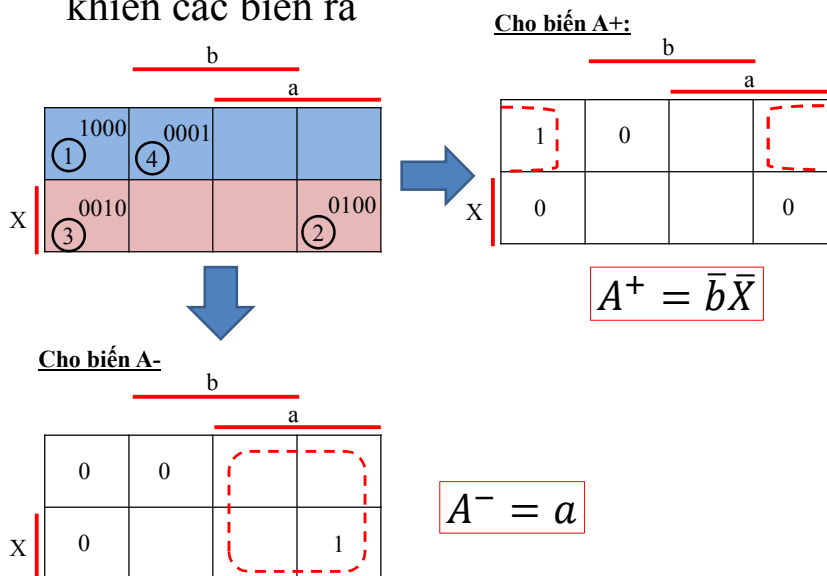
- Xác định và mã hóa biến trung gian:
  - $S_{min} = 1$ , chọn biến trung gian là biến ra X (không thể lấy biến ra là biến trung gian)



45

45

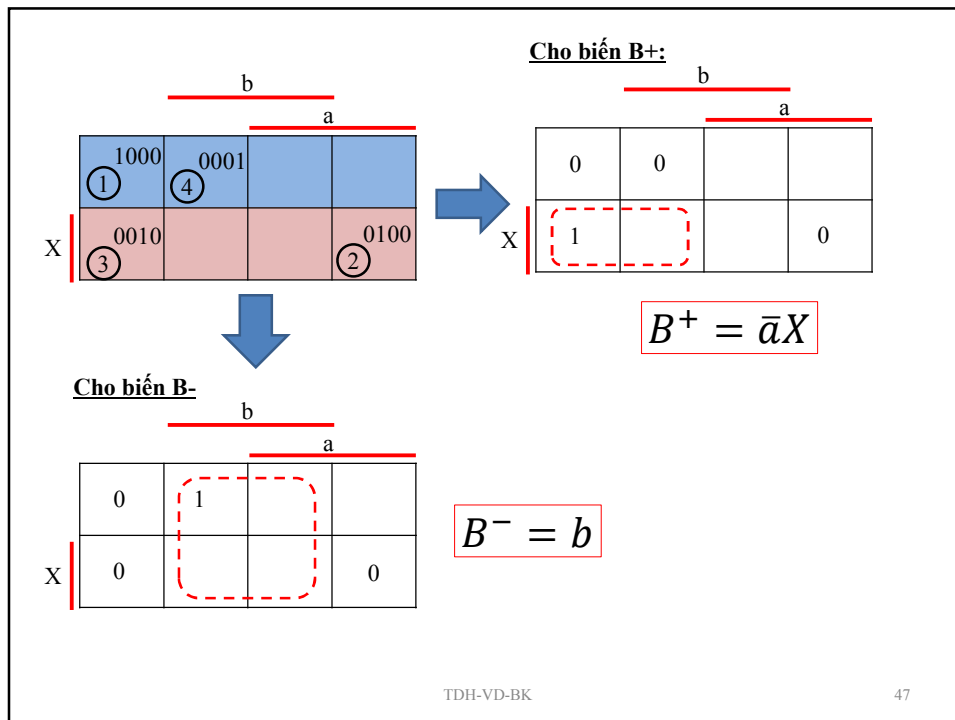
- Lập bảng Các nô để xác định hàm logic điều khiển các biến ra



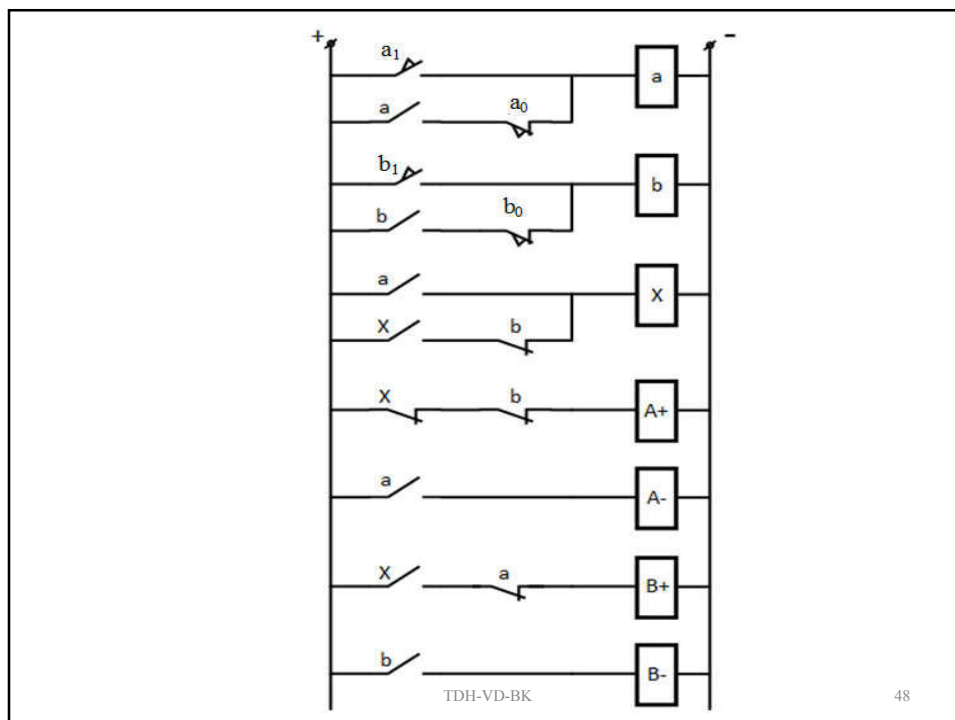
TDH-VD-BK

46

46



47



48

## 3. Tổng hợp mạch logic tuần tự

### 3.1. Khái niệm mạch logic tuần tự

- Định nghĩa
- Tính chất
- Phân loại
- Biểu diễn bằng đồ thị thời gian

### 3.2. Tổng hợp mạch logic tuần tự

- Phương pháp ma trận trạng thái
- Phương pháp GRAFCET

TDH-VD-BK

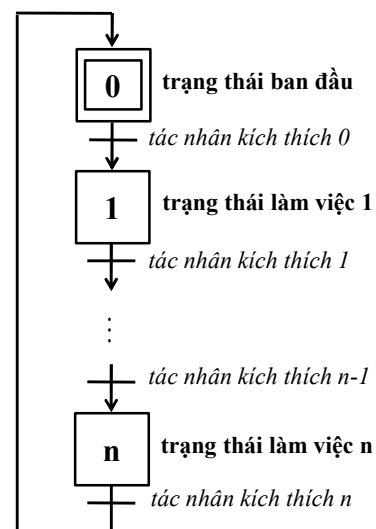
49

49

## 3.2. Tổng hợp mạch logic tuần tự

### • Phương pháp GRAFCET

- Biểu diễn các quá trình công nghệ dưới dạng lưu đồ (graph) các trạng thái làm việc
- Xây dựng các hàm logic điều khiển và sơ đồ điều khiển từ lưu đồ (graph) các trạng thái làm việc



TDH-VD-BK






50

50



- Phương pháp GRAFCET

- Một số ký hiệu cơ bản

Tên gọi	Ký hiệu
Trạng thái ban đầu	
Trạng thái thông thường	
Trạng thái đang hoạt động	 
Cung định hướng và chuyển tiếp	

TDH-VD-BK

51

51

- Phương pháp GRAFCET

- Mỗi trạng thái ứng với một hoặc một nhóm hành động hoàn chỉnh
- Mỗi chuyển tiếp đi kèm với tác nhân kích thích (điều kiện logic) biểu thị điều kiện chuyển trạng thái
- Trạng thái đang hoạt động: thực thi các hành động tương ứng với trạng thái đó
- Hoạt động của GRAFCET: các trạng thái lần lượt hoạt động theo trình tự quy định (di chuyển token)

TDH-VD-BK

52

52

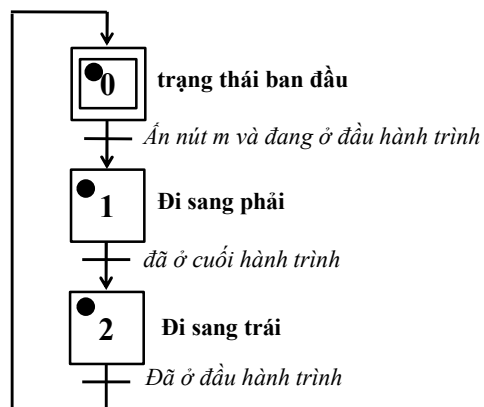
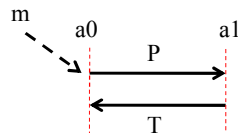
- Quy tắc hoạt động của GRAFCET  
(quy tắc vượt qua chuyển tiếp)
  - Chuyển tiếp sẵn sàng: các trạng thái ngay trước chuyển tiếp (đầu vào) là đang hoạt động
  - Chuyển tiếp được vượt qua: khi chuyển tiếp sẵn sàng và tác nhân kích thích xảy ra (điều kiện logic là đúng)
  - Khi vượt qua chuyển tiếp: Các trạng thái ngay trước chuyển tiếp ngừng hoạt động, đồng thời các trạng thái ngay sau (đầu ra) hoạt động

TDH-VD-BK

53

53

Ví dụ 1:



TDH-VD-BK

54

54

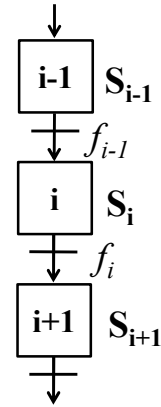
- Xây dựng hàm logic từ GRAFCET

- Mỗi trạng thái  $i$  ứng với một biến ra  $S_i$
- Mỗi biến  $S_i$  sẽ có 2 hàm đóng (set)  $S_i^+$  và hàm cắt (reset)  $S_i^-$

$$\begin{aligned} S_i &= (S_i^+ + S_i^-) \overline{S_i} \\ S_i^+ &= f_{i-1} S_{i-1} \\ S_i^- &= S_{i+1} \end{aligned}$$

- Cần một tín hiệu xác lập trạng thái ban đầu ( $g$ )

$$S_0^+ = g + f_n S_n$$

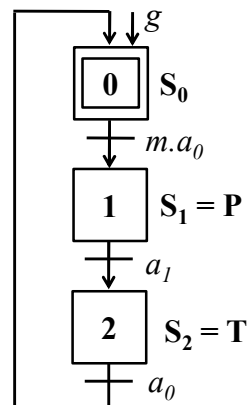
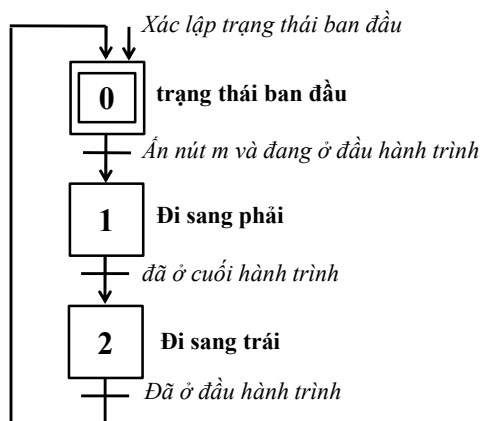
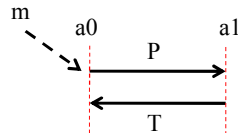


TDH-VD-BK

55

55

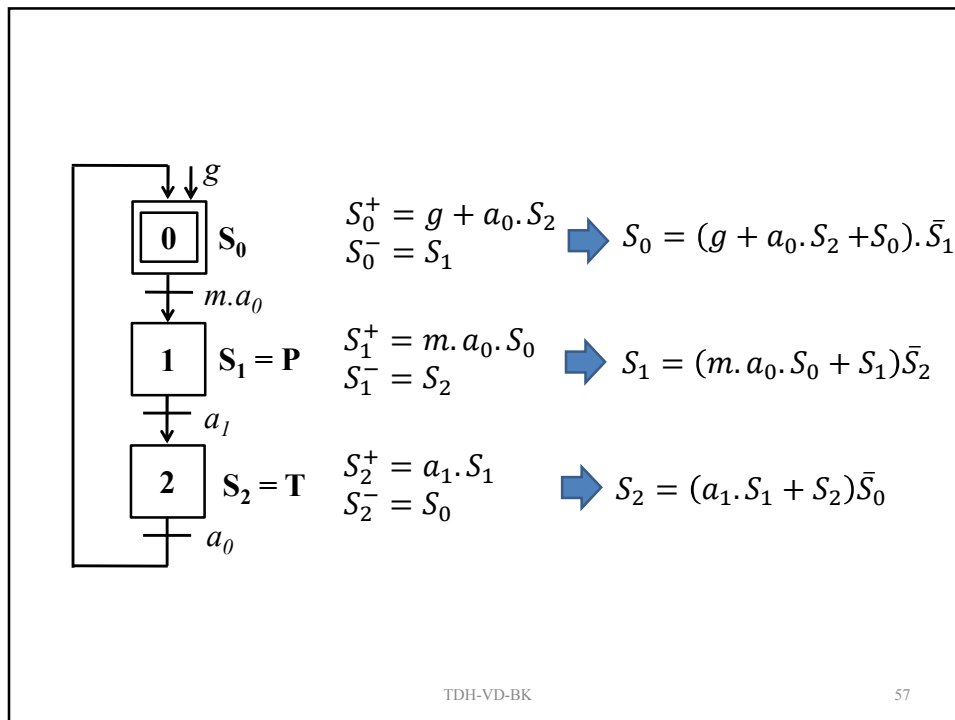
Ví dụ 2:



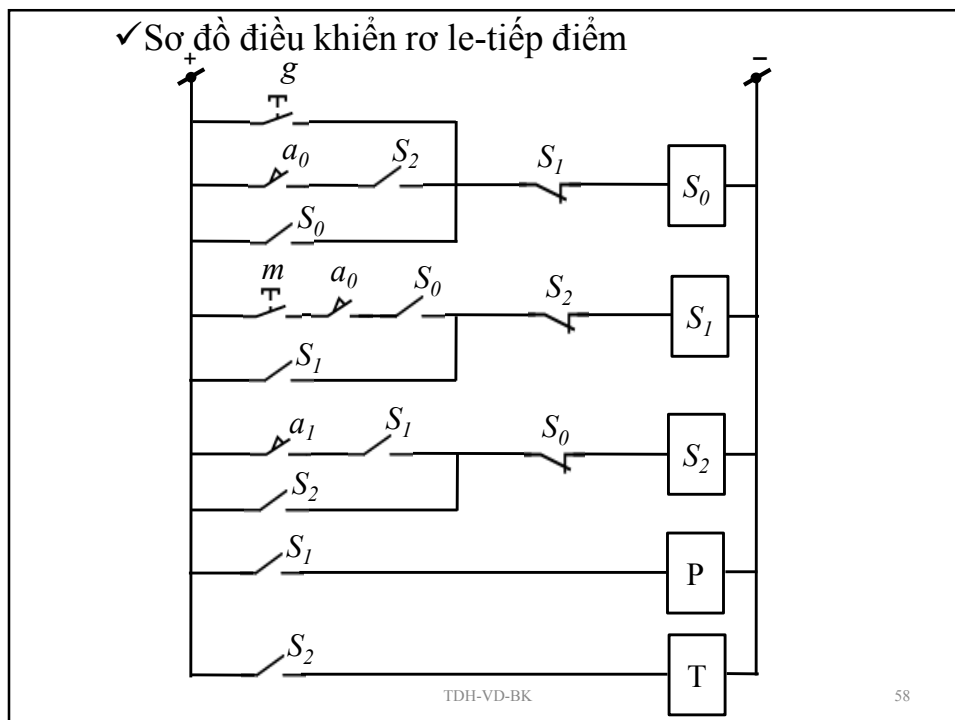
TDH-VD-BK

56

56



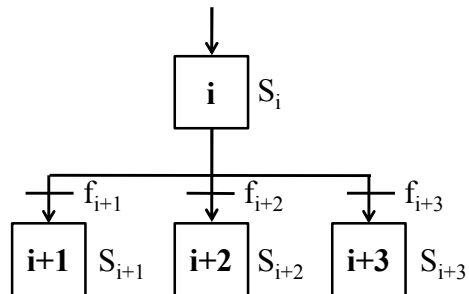
57



58

– Các dạng mạch đặc biệt

- Mạch phân kỳ “HOẶC”



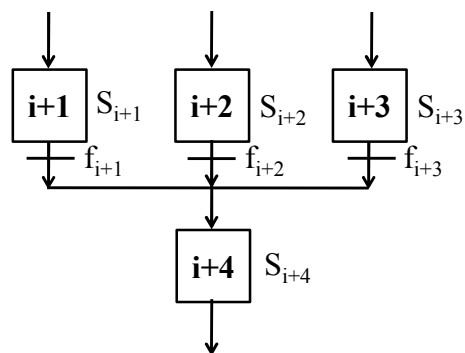
$$\begin{aligned} S_i^- &= S_{i+1} + S_{i+2} + S_{i+3} \\ S_{i+1}^+ &= f_{i+1} S_i \\ S_{i+2}^+ &= f_{i+2} S_i \\ S_{i+3}^+ &= f_{i+3} S_i \end{aligned}$$

TDH-VD-BK

59

59

- Mạch hội tụ “HOẶC”



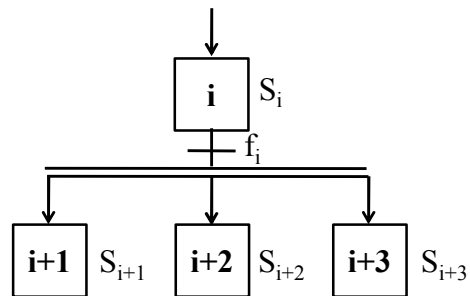
$$\begin{aligned} S_{i+1}^- &= S_{i+2}^- = S_{i+3}^- = S_{i+4} \\ S_{i+4}^+ &= f_{i+1} S_{i+1} + f_{i+2} S_{i+2} + f_{i+3} S_{i+3} \end{aligned}$$

TDH-VD-BK

60

60

• Mạch phân kỳ “VÀ”



$$S_i^- = S_{i+1} \cdot S_{i+2} \cdot S_{i+3}$$

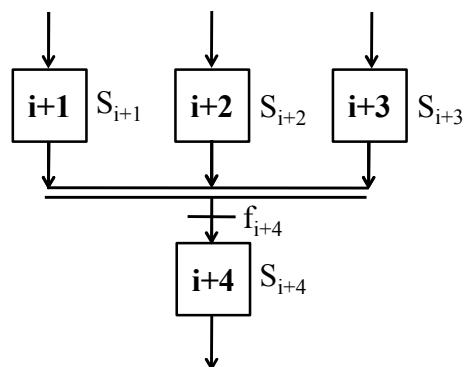
$$S_{i+1}^+ = S_{i+2}^+ = S_{i+3}^+ = f_i \cdot S_i$$

TDH-VD-BK

61

61

• Mạch hội tụ “VÀ”



$$S_{i+1}^- = S_{i+2}^- = S_{i+3}^- = S_{i+4}$$

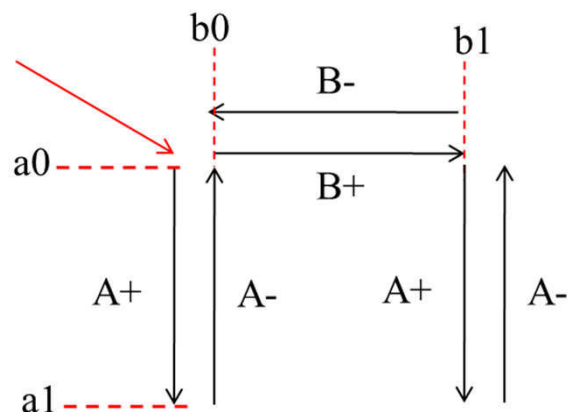
$$S_{i+4}^+ = f_{i+4} \cdot S_{i+1} \cdot S_{i+2} \cdot S_{i+3}$$

TDH-VD-BK

62

62

• Ví dụ 3



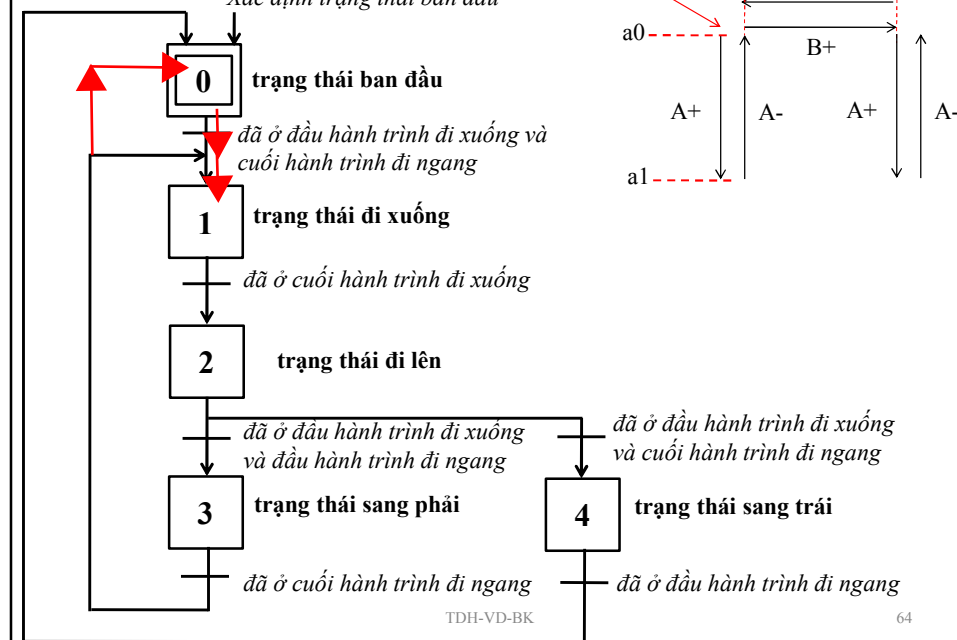
TDH-VD-BK

63

63

• Ví dụ 3

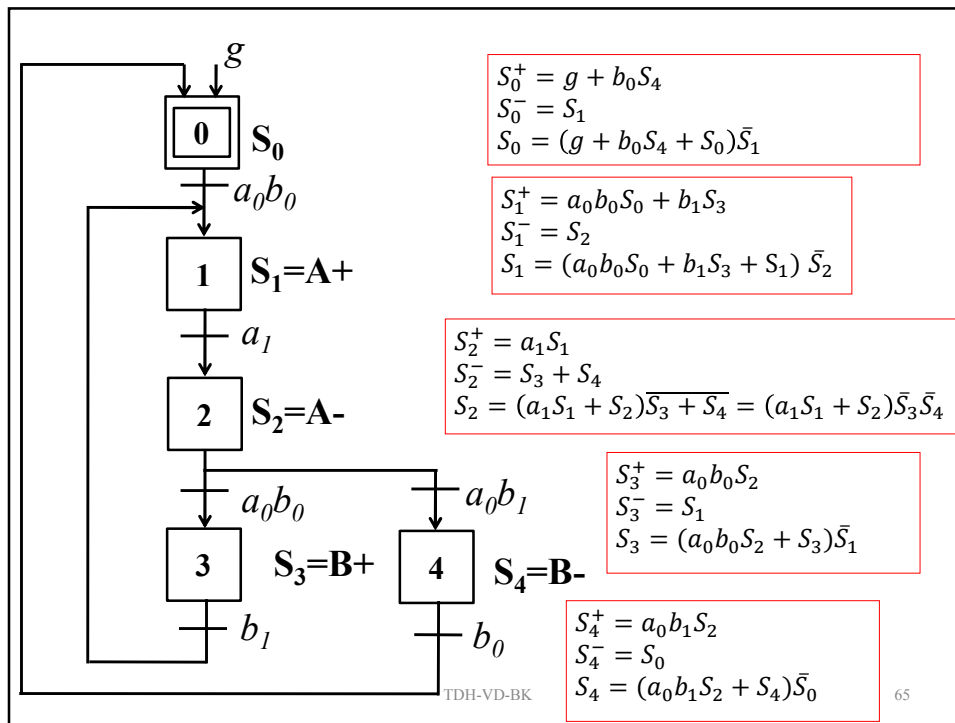
Xác định trạng thái ban đầu



TDH-VD-BK

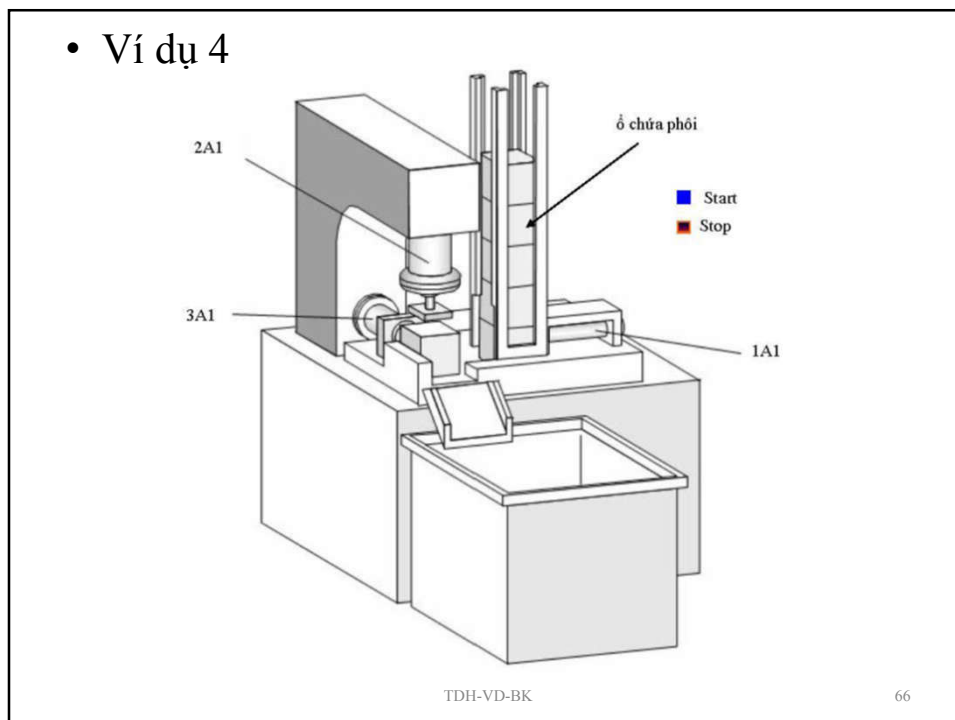
64

64



65

#### • Ví dụ 4



66



# ĐIỀU KHIỂN LOGIC VÀ PLC

TDH-VD-BK

1

1

## Nội dung

1. Cơ sở cho Điều khiển logic
2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp
3. Tổng hợp mạch logic tuần tự
- 4. Tổng quan về PLC**
5. Kỹ thuật lập trình PLC

TDH-VD-BK

2

2

## 4. Tổng quan về PLC

### 4.1. Giới thiệu chung về PLC

- Định nghĩa
- Lịch sử
- Ưu điểm
- Ứng dụng
- Phân loại

### 4.2. Cấu trúc phần cứng và nguyên lý làm việc

### 4.3. Ghép nối với module vào/ra logic

### 4.4. Ghép nối với module vào/ra tương tự

### 4.5. Chuẩn IEC 61131-3 cho lập trình PLC

TDH-VD-BK

3

3

## 4.1. Giới thiệu chung về PLC

- Định nghĩa (theo IEC61131):

*“Hệ thống điện tử số được thiết kế sử dụng trong môi trường công nghiệp, có bộ nhớ khả trình với tập lệnh hướng tới người sử dụng để thực hiện các chức năng nhất định như logic, tuần tự, định thời gian, đếm và tính toán số học, được sử dụng để điều khiển nhiều loại máy và quá trình khác nhau thông qua các đầu vào/ra số hoặc tương tự.”*

TDH-VD-BK

4

4

## 4.1. Giới thiệu chung về PLC

- Lịch sử
  - ✓ Xuất hiện năm 1968 bởi Modicon theo yêu cầu của General Motors
  - ✓ Được thiết kế để dễ lập trình và bảo trì.
  - ✓ Thay thế các hệ thống điều khiển logic công kênh sử dụng rơle trong sản xuất tự động.
  - ✓ Chi phí thấp, nhỏ gọn, linh hoạt dựa trên kiến trúc bộ vi xử lý.



TDH-VD-BK

5

5

## 4.1. Giới thiệu chung về PLC

- Ưu điểm:
  - ✓ Chi phí hiệu quả.
  - ✓ Linh hoạt.
  - ✓ Cho phép điều khiển các tác vụ phức tạp nhờ khả năng tính toán đa dạng.
  - ✓ Lập trình đơn giản.
  - ✓ Tin cậy.

TDH-VD-BK

6

6

## 4.1. Giới thiệu chung về PLC

- Ứng dụng của PLC:
  - ✓ Điều khiển hệ thống băng tải, điều khiển thang máy, hệ thống quản lý và giám sát, hệ thống điều hòa.
  - ✓ Quản lý hệ điều khiển phân tán phức tạp
  - ✓ Hệ thống điều khiển trong nhà máy xi măng, nhà máy nhiệt điện.
  - ✓ ...

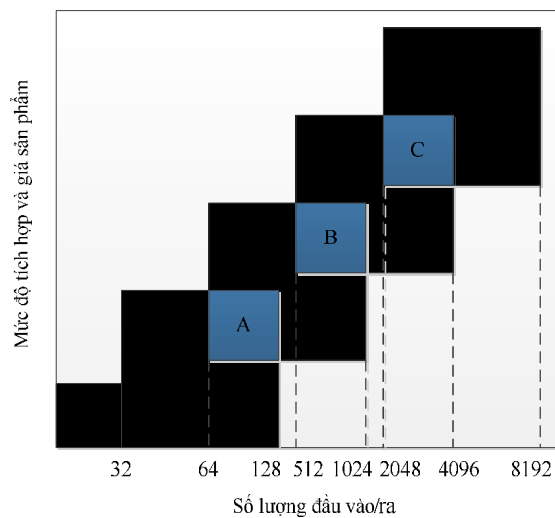
TDH-VD-BK

7

7

## 4.1. Giới thiệu chung về PLC

- Phân loại



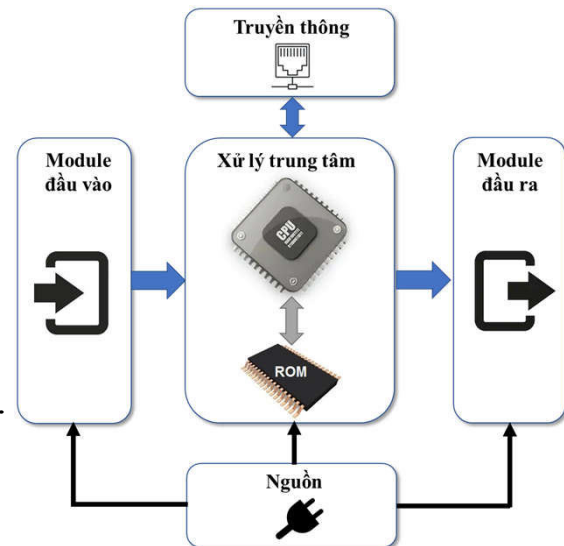
TDH-VD-BK

8

8

## 4.2. Cấu trúc phần cứng và nguyên lý làm việc

- Cấu trúc phần cứng
  - ✓ Nguồn (Power Supply).
  - ✓ Bộ xử lý trung tâm (CPU).
  - ✓ Đầu vào/ra (I/O).
  - ✓ Bộ nhớ (Memory).
  - ✓ Truyền thông (Communication).



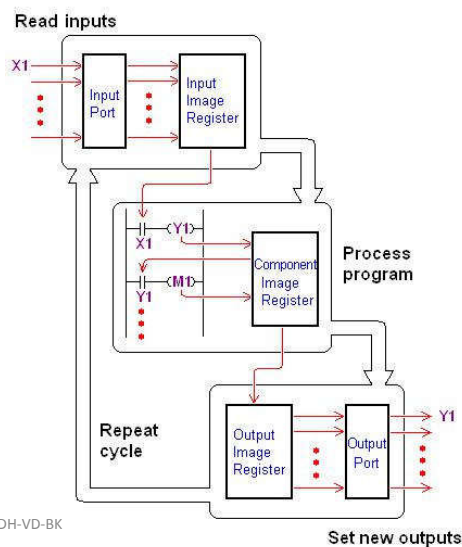
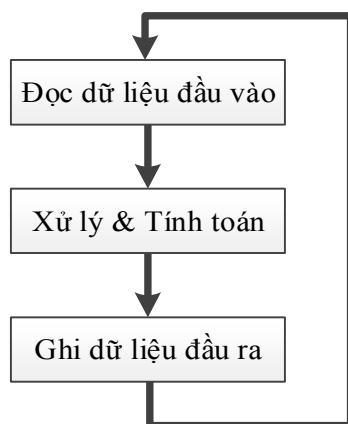
TDH-VD-BK

9

9

## 4.2. Cấu trúc phần cứng và nguyên lý làm việc

- Nguyên lý làm việc: theo chu trình quét



TDH-VD-BK

10

10

## 4.2. Ghép nối với module vào/ra

- Thiết bị đầu vào logic



Nút ấn



Công tắc chuyển mạch



Công tắc hành trình



Cảm biến điện dung

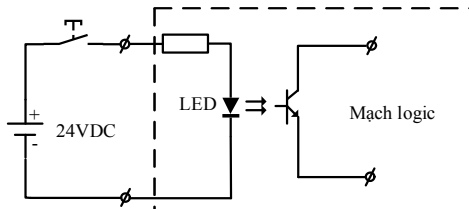
TDH-VD-BK

11

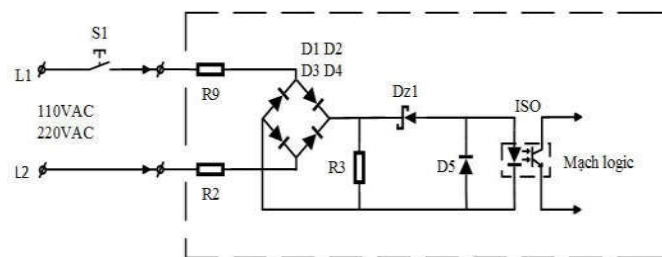
11

## 4.2. Ghép nối với module vào/ra

- Sơ đồ nguyên lý ghép nối đầu vào logic



Sơ đồ nguyên lý mạch module đầu vào 24VDC



Sơ đồ nguyên lý mạch module đầu vào 110VAC – 220VAC

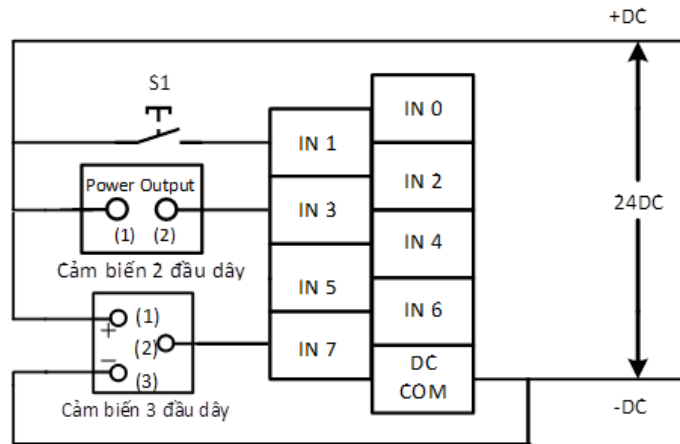
TDH-VD-BK

12

12

## 4.2. Ghép nối với module vào/ra

- Ví dụ



Sơ đồ nguyên lý đấu dây với module  
1769-IQ của hãng Rockwell Automation  
TDH-VD-BK

13

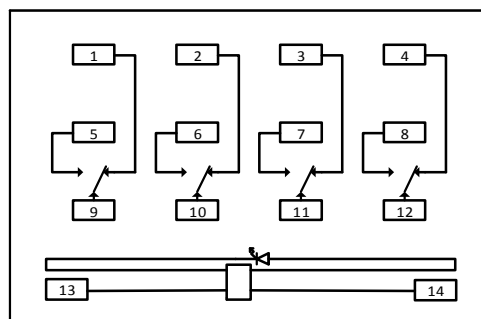
13

## 4.2. Ghép nối với module vào/ra

- Thiết bị đầu ra logic



Role



TDH-VD-BK



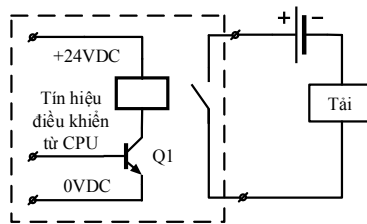
Van điện từ

14

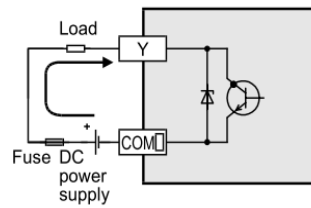
14

## 4.2. Ghép nối với module vào/ra

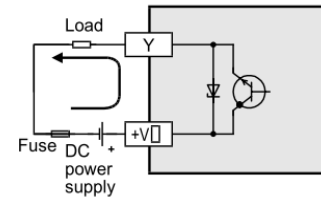
- Sơ đồ nguyên lý ghép nối đầu ra logic



Đầu ra kiểu role



Đầu ra transistor kiểu sink



Đầu ra transistor kiểu source

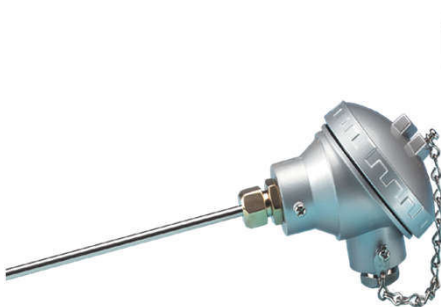
TDH-VD-BK

15

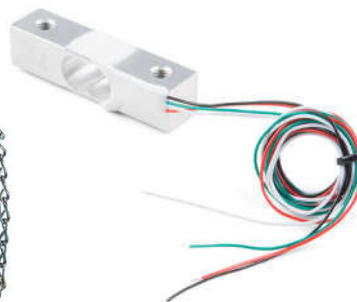
15

## 4.2. Ghép nối với module vào/ra

- Thiết bị đầu vào tương tự



Cảm biến nhiệt độ



Cảm biến trọng lượng (loadcell)



Cảm biến mức

TDH-VD-BK

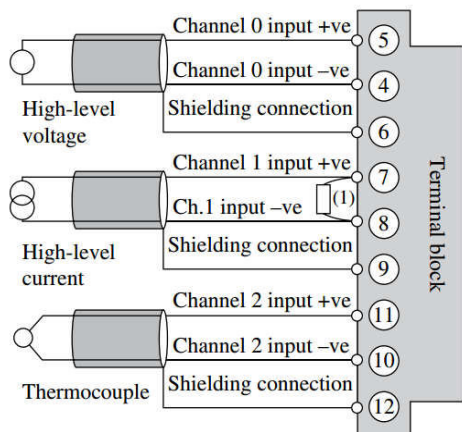
16

16

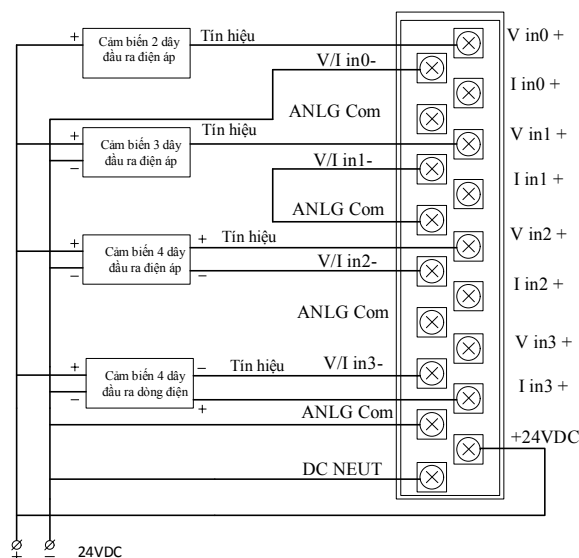


## 4.2. Ghép nối với module vào/ra

- Ghép nối đầu vào tương tự



Kết nối cảm biến sử dụng cáp có vỏ chống nhiễu



Kết nối cảm biến 2,3,4 dây với module 1769-IF4

TDH-VD-BK

hãng Rockwell Automation

17

17

## 4.2. Ghép nối với module vào/ra

- Thiết bị đầu ra tương tự



Van điện



Biến tần

TDH-VD-BK

18

18

## 4.2. Ghép nối với module vào/ra

- Ghép nối đầu ra tương tự



19

## 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Tiêu chuẩn IEC 61131
- Ưu điểm của IEC61131-3 đối với người lập trình.
- Các mô hình phần mềm trong IEC 61131-3
- 5 ngôn ngữ lập trình trong IEC 61131-3
- Khối chức năng và POU
- Kiểu dữ liệu IEC, hàm tiêu chuẩn và khối chức năng tiêu chuẩn
- Chứng nhận PLCopen

TDH-VD-BK

20

20

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Tiêu chuẩn IEC 61131

Phần 1	Tổng quan chung, các định nghĩa	Tiêu chuẩn quốc tế
Phần 2	Phần cứng	Tiêu chuẩn quốc tế
<b>Phần 3</b>	<b>Ngôn ngữ lập trình</b>	<b>Tiêu chuẩn quốc tế</b>
Phần 4	Hướng dẫn người dùng	Tiêu chuẩn quốc tế
Phần 5	Truyền thông	Tiêu chuẩn quốc tế
Phần 6	An toàn	Tiêu chuẩn quốc tế
Phần 7	Logix mở	Tiêu chuẩn quốc tế
Phần 8	Báo cáo kỹ thuật	

TDH-VD-BK

21

21

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

#### **Phần 3: IEC 61131-3 Ngôn ngữ lập trình**

- ✓ Mô hình phần mềm, truyền thông và lập trình.
- ✓ Định nghĩa 5 ngôn ngữ lập trình:
  - Cú pháp và ngữ nghĩa của hai ngôn ngữ văn bản và hai ngôn ngữ đồ họa: Instruction List (IL), Structured Text (ST), Ladder Diagram (LD) và Function Block Diagram (FBD).
  - Sequential Function Chart (SFC) cho lập trình cấu trúc.

TDH-VD-BK

22

22

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Ưu điểm của IEC 61131 – 3
  - Tiêu chuẩn quốc tế.
    - ✓ Từng bước được hỗ trợ bởi các nhà sản xuất
    - ✓ Cấu trúc, ngôn ngữ và cách xử lý thống nhất
  - Tiết kiệm thời gian
    - ✓ Thống nhất các khái niệm về mô hình phần mềm và kiểu dữ liệu
    - ✓ Học và áp dụng cho nhiều bộ điều khiển
    - ✓ Giảm thiểu hiểm lẫn và sai sót.
    - ✓ Hàm và khối chức năng chuẩn
    - ✓ Có tính tái sử dụng

TDH-VD-BK

23

23

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Ưu điểm của IEC 61131 – 3 (tiếp)
  - Hỗ trợ lập trình àn toàn và chất lượng.
    - ✓ Cấu trúc đơn giản và thuận tiện
    - ✓ Tạo dữ liệu hạn chế lỗi lập trình
  - Cung cấp ngôn ngữ lập trình tốt nhất cho từng vấn đề
    - ✓ Thông số kỹ thuật thống nhất cho các ngôn ngữ
    - ✓ Hai ngôn ngữ văn bản và hai ngôn ngữ đồ họa.
    - ✓ Một ngôn ngữ cấu trúc, cung cấp một cái nhìn tổng quan.
    - ✓ Ngôn ngữ bậc cao khả dụng
    - ✓ Có khả năng kết hợp các ngôn ngữ

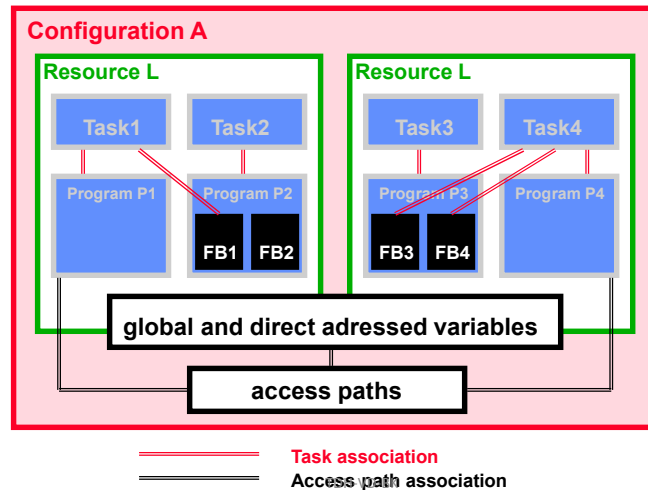
TDH-VD-BK

24

24

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Mô hình phần mềm



25

25

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Mô hình phần mềm:
  - ✓ Configuration - Cấu hình: Tương đương với một hệ thống điều khiển khả trình.
  - ✓ Resource – Tài nguyên: Tương ứng với chức năng xử lý tín hiệu với giao diện sử dụng người-máy và chức năng kết nối với cảm biến và cơ cấu chấp hành
  - ✓ Task – Tác vụ: cách thức thực thi các đơn vị tổ chức chương trình theo khoảng thời gian định kỳ hay theo sự kiện

TDH-VD-BK

26

26

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Mô hình phần mềm (tiếp)
  - ✓ Program - Chương trình: đơn vị tổ chức chương trình cấp cao nhất, có thể gọi đến hàm hoặc khối chức năng.
  - ✓ Function Block – Khối chức năng: đơn vị tổ chức chương trình, có thể gọi đến hàm hoặc khối chức năng khác

TDH-VD-BK

27

27

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Các thành phần

Thành phần	Ví dụ
Character set	English, ...
Data types	BOOL, WORD, INTERGER, ...
Variables	VAR, VAR_input, VAR_output, ...
Functions	AND, OR, ADD, SUB, ...
Program Organisation Units (POUs)	Function, Function Block, ...
SFC elements	Steps, Transitions, ...
Configuration elements	Tasks

TDH-VD-BK

28

28

## 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- Một số kiểu dữ liệu cơ bản

STT	Từ khóa	Kiểu dữ liệu	Bit dữ liệu	Giá trị khởi tạo mặc định
1	BOOL	Kiểu Bool	1	0/FALSE
2	INT	Kiểu nguyên	16	0
3	REAL	Số thực	32	0.0
4	TIME	Khoảng thời gian	--	T#0s
5	DATE	Ngày	--	D#0001-01-01
6	TIME_OF_DAY (TOD)	Thời gian trong ngày	--	TOD#00:00:00
7	STRING	Chuỗi kí tự 8 bit	8	' '
8	BYTE	Chuỗi 8 bit	8	0
9	WORD	Chuỗi 16 bit	16	0
10	DWORD	Chuỗi 32 bit	32	0

TDH-VD-BK

29

29

## 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

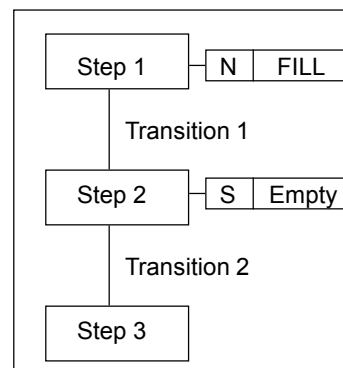
Instruction List

```
LD    A
ANDN  B
ST    C
```

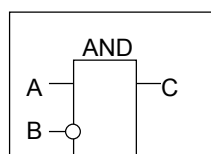
Structured Text

```
C:= A AND NOT B
```

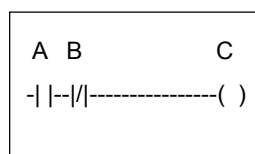
Sequential Function Chart



Function Block Diagram



Ladder Diagram



TDH-VD-BK

30

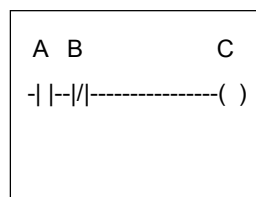
30

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### **Ladder Diagram (LD)**

- ✓ Chuẩn hóa các ký hiệu lập trình giản đồ thang dạng tiếp điểm.
- ✓ Dựa trên tiêu chuẩn cho bản vẽ điện



TDH-VD-BK

31

31

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### **Instruction List (IL)**

- ✓ Mô hình thực thi dựa trên tính toán đơn.
- ✓ Chỉ một thao tác như lưu trữ một giá trị trong thanh ghi tích lũy, được phép trên mỗi dòng lệnh

LD	A
ANDN	B
ST	C

TDH-VD-BK

32

32



### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### **Structured Text (ST)**

- ✓ Ngôn ngữ lập trình bậc cao.
- ✓ Cú pháp tương tự PASCAL
- ✓ Hỗ trợ các câu lệnh phức tạp: REPEAT-UNTIL; WHILE-DO; IF-THEN-ELSE; CASE; SQR(), SIN()

C:= A AND NOT B

TDH-VD-BK

33

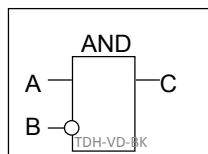
33

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### **Function Block Diagram (ST)**

- ✓ Ngôn ngữ đồ họa, được sử dụng phổ biến ở Châu Âu.
- ✓ Các phần tử chương trình xuất hiện dưới dạng các khối được nối dây với nhau tương tự như sơ đồ mạch.
- ✓ Được sử dụng trong nhiều ứng dụng liên quan đến luồng thông tin hoặc dữ liệu giữa các thành phần điều khiển.



34

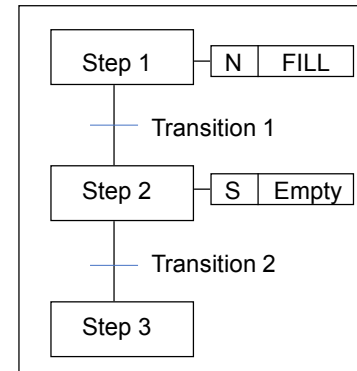
34

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### Sequential Function Chart (SFC)

- ✓ Ngôn ngữ đồ họa dạng biểu đồ tuần tự --> flowchart.
- ✓ Dựa trên Grafcet (IEC 848)
- ✓ Phù hợp cho chẩn đoán nhanh
- ✓ Các thành phần cơ bản gồm các trạng thái (Step) đi kèm khối hành động và các chuyển tiếp (Transition)



TDH-VD-BK

35

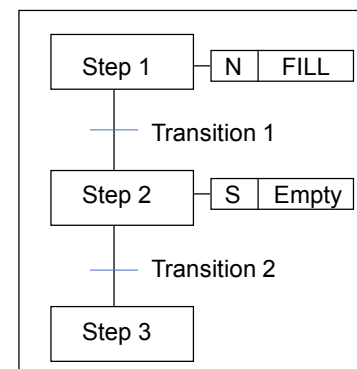
35

### 4.3. Chuẩn IEC 61131-3 cho lập trình PLC

- 5 ngôn ngữ lập trình

#### Sequential Function Chart (SFC)

- ✓ Các trạng thái bao gồm một đoạn chương trình được thực thi cho đến khi điều kiện tại chuyển tiếp được thỏa mãn.
- ✓ Lập trình các tác vụ phức tạp bằng cách chia thành các phần nhỏ hơn
- ✓ Mỗi thành phần có thể được lập trình bằng bất kỳ ngôn ngữ nào



TDH-VD-BK

36

36

# ĐIỀU KHIỂN LOGIC VÀ PLC

TDH-VD-BK

1

1

## Nội dung

1. Cơ sở cho Điều khiển logic
2. Tổng hợp và tối thiểu hóa mạch logic tổ hợp
3. Tổng hợp mạch logic tuần tự
4. Tổng quan về PLC
- 5. Kỹ thuật lập trình PLC**

TDH-VD-BK

2

2

## 5. Kỹ thuật lập trình PLC

- 5.1. Chu trình thiết kế chương trình PLC
- 5.2. Các ngôn ngữ lập trình theo chuẩn IEC61131-3
- 5.3. Thiết kế chương trình sử dụng hàm logic
- 5.4. Thiết kế chương trình sử dụng SFC

TDH-VD-BK

3

3

## 5. Kỹ thuật lập trình PLC

- 5.1. Chu trình thiết kế chương trình PLC**
- 5.2. Các ngôn ngữ lập trình theo chuẩn IEC61131-3
- 5.3. Thiết kế chương trình sử dụng hàm logic
- 5.4. Thiết kế chương trình sử dụng SFC

TDH-VD-BK

4

4

## 5.1. Chu trình thiết kế chương trình PLC

Bước 1 Phân tích

Bước 2 Thiết kế

Bước 3 Lập trình

Bước 4 Kiểm tra

Bước 5 Viết tài liệu

Bước 6 Vận hành

TDH-VD-BK

5

5

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 1: Phân tích**

- Tìm hiểu công nghệ
  - ✓ Thảo luận trực tiếp với khách hàng
  - ✓ Sơ đồ công nghệ P&ID, hồ sơ nâng cấp cải tạo (nếu có)
- Trả lời các câu hỏi
  - ✓ Cần điều khiển những gì?
  - ✓ Các hành động điều khiển được thực hiện như thế nào
  - ✓ Người vận hành tác động được những gì?
  - ✓ Xử lý thế nào khi có sự cố?

6

6

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 1: Phân tích (tiếp)**

- Kết quả
  - ✓ Mô tả được hoạt động của quá trình
  - ✓ Tác động từ trạm vận hành và cách thức tác động
  - ✓ Danh sách các tín hiệu vào/ra
  - ✓ Chế độ vận hành khi có lỗi

TDH-VD-BK

7

7

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 2: Thiết kế**

- Lựa chọn phần cứng
  - ✓ Số lượng đầu vào/ra logic hoặc tương tự
  - ✓ Số lượng đầu vào/ra đặc biệt: high speed counter, PWM
  - ✓ Truyền thông: RS232, Modbus, Ethernet...

STT	Tên tín hiệu	Đầu vào		Đầu ra		Truyền thông hoặc tín hiệu đặc biệt		Ghi chú
		Logic	Tương tự	Logic	Tương tự	Loại truyền thông	Loại tín hiệu	
1								
...								

TDH-VD-BK

8

8

## 5.1. Chu trình thiết kế chương trình PLC

### • **Bước 2: Thiết kế (tiếp)**

- Lựa chọn phần cứng
  - ✓ Tốc độ xử lý, dung lượng bộ nhớ

STT	Loại câu lệnh	Thời gian max ( $\mu$ s)	Thời gian min ( $\mu$ s)	Bộ nhớ chương trình (Words)	Bộ nhớ dữ liệu (Words)	Số lượng câu lệnh (number)	Tổng bộ nhớ chương trình (Words)
1	Đọc đầu vào						

TDH-VD-BK

9

9

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 2: Thiết kế (tiếp)**

- Phân địa chỉ tín hiệu
  - ✓ Theo chức năng: đầu vào/ra tương tự, tốc độ cao...
  - ✓ Các biến có cùng một đối tượng hoặc các biến trong cùng trình tự tác động nên có địa chỉ gần nhau

STT	Tên tín hiệu	Đầu vào		Đầu ra		Địa chỉ	Ghi chú
		Logic	Tương tự	Logic	Tương tự		
1							
2							
...							

TDH-VD-BK

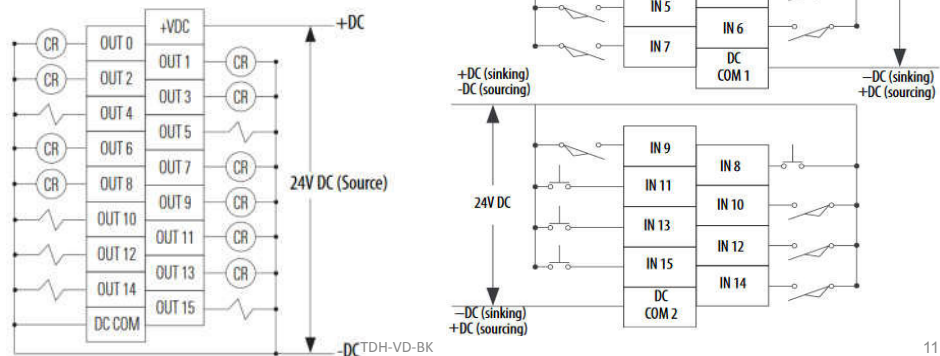
10

10

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 2: Thiết kế (tiếp)**

- Phân địa chỉ tín hiệu
  - ✓ Sơ đồ đấu dây



11

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 3: Lập trình**

- Tổng hợp hàm logic --> chương trình LD, IL, FBD
- Grafcet --> chương trình SFC
- Lưu đồ thuật toán --> chương trình ST

TDH-VD-BK

12

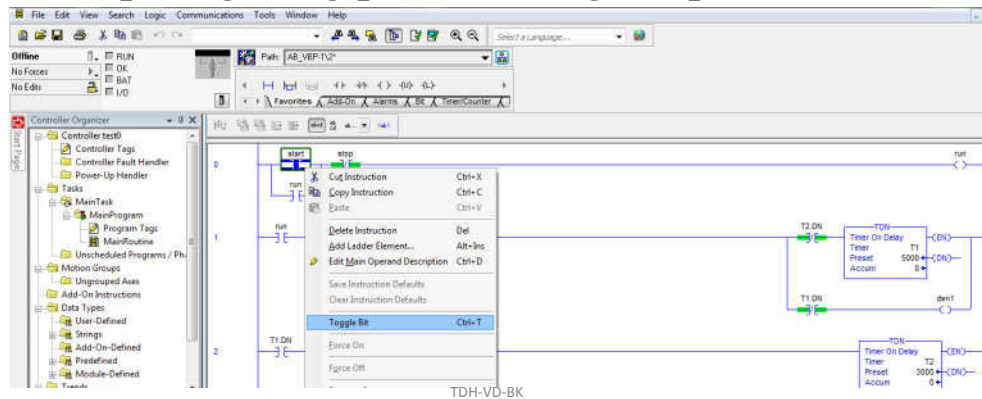
12



## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 4: Kiểm tra**

- Mô phỏng bằng phần mềm, giả lập các tín hiệu vào



13

13

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 5: Viết tài liệu**

- Hướng dẫn vận hành
- Phục vụ bảo trì, bảo dưỡng
- Viết song song và được cập nhật trong quá trình phát triển dự án

#### 2.3.2. PLC Communication

The PLC communicates with

- HMI by [Profinet](#).
- Each VFD of conveyor by [Profinet](#).
- UHT PLC by [Profinet](#).
- Each Stopper, Sensor by Hardwire (Inputs/Outputs on the ASI Network)

For the Communication table between PLC and machines, refer to [802ST3 Conveyor 012 A](#)

#### 2.3.3. PLC Inputs/Outputs list

For the inputs and outputs list, refer to the file [802ST3 Conveyor 011 A](#)

#### 2.3.4. Defaults list

For the defaults list, refer to the file [802ST3 Conveyor 010 A](#)

#### 3. DETAILED PROCESS SYSTEM DESCRIPTION

All the start-up, running and stop requirements for each function are listed in the different function descriptions.

This chapter summarizes:

- The aim of each function or each operation,
- The document number of the function description.

#### 3.1. Transfer empty boxes from Box Dispenser Room to [Corazza 1](#)

TDH-VD-BK

14

14

## 5.1. Chu trình thiết kế chương trình PLC

### **Bước 6: Vận hành**

- Chạy thử bộ từng phận riêng lẻ, không tải
- Chạy thử từng bộ phận riêng lẻ, có tải
- Chạy thử toàn hệ thống không tải
- Chạy thử toàn hệ thống có tải

TDH-VD-BK

15

15

## 5. Kỹ thuật lập trình PLC

5.1. Chu trình thiết kế chương trình PLC

### **5.2. Các ngôn ngữ lập trình**

5.3. Thiết kế chương trình sử dụng hàm logic

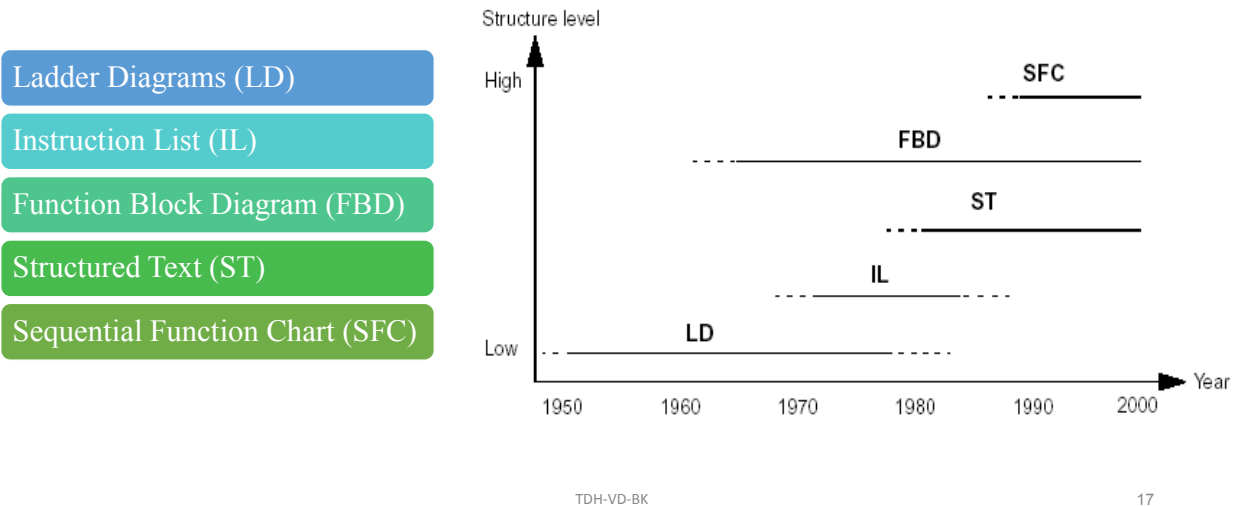
5.4. Thiết kế chương trình sử dụng SFC

TDH-VD-BK

16

16

## 5.2. Các ngôn ngữ lập trình



17

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ LD

- Hai thanh nguồn (Power rails).
- Tiếp điểm (Contacts) đại diện cho biến logic
  - ✓ Thường mở (Normally Open)
  - ✓ Thường đóng (Normally Closed)
  - ✓ Tiếp điểm nối tiếp --> logic AND
  - ✓ Nhánh song song --> logic OR
- Cuộn dây (Coils) đại diện cho đầu ra
- Tạo mạch phản hồi: tên tiếp điểm trùng tên cuộn dây

TDH-VD-BK

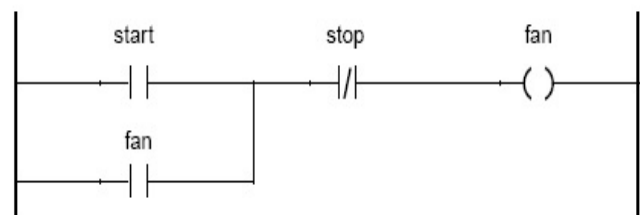
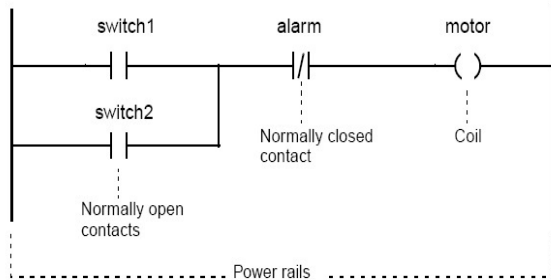
18

18

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ LD

*Motor := (switch1 OR switch 2) AND (NOT alarm)*



*fan := (start OR fan) AND (NOT stop)*

TDH-VD-BK

19

19

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ LD

STT	Tên	Ký hiệu	Ý nghĩa
1	Tiếp điểm thường hở		Khi biến có giá trị bằng 1 thì tiếp điểm này sẽ đóng lại.
2	Tiếp điểm thường kín		Khi biến có giá trị bằng 0 thì tiếp điểm này sẽ đóng lại.
3	Tiếp điểm sườn lên		Phát hiện trạng thái của biến thay đổi từ 0 lên 1 và sẽ cho giá trị 1 tại thời điểm đó. Còn các thời điểm khác là 0.
4	Tiếp điểm sườn xuống		Phát hiện trạng thái của biến thay đổi từ 1 xuống 0 và sẽ cho giá trị 1 tại thời điểm đó. Còn các thời điểm khác là 0.

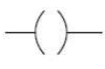
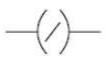
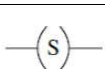

TDH-VD-BK

20

20

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ LD

5	Cuộn dây		Trạng thái của biến tương ứng với trạng thái bên trái cuộn dây.
6	Cuộn dây đảo		Trạng thái của biến tương ứng với nghịch đảo của trạng thái bên trái cuộn dây.
7	Cuộn dây SET		Trạng thái của biến tương ứng được set lên ON khi trạng thái ON ở phía trước và duy trì trạng thái này cho đến khi được reset bằng cuộn RESET.
8	Cuộn dây RESET		Trạng thái của biến tương ứng được reset xuống OFF khi có trạng thái ON phía trước.

TDH-VD-BK

21

21

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ LD

- Ưu điểm:
  - ✓ Dễ lập trình, dễ hiểu: giống sơ đồ mạch điện.
  - ✓ Dễ bảo dưỡng: có khả năng chẩn đoán lỗi online, từ đó định vị lỗi logic hoặc lỗi thiết bị.
- Nhược điểm:
  - ✓ Khó module hóa.
  - ✓ Hạn chế với kiểu dữ liệu có cấu trúc.

TDH-VD-BK

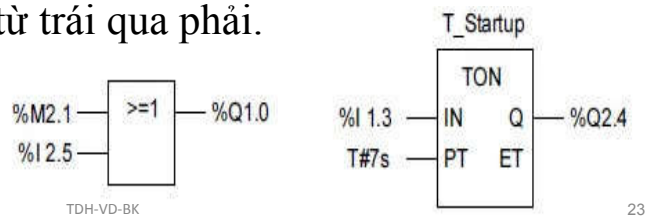
22

22

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ FBD

- Gồm nhiều Function Block (FB).
- Tín hiệu chạy từ đầu ra của FB này đến đầu vào của FB khác.
- Đầu ra FB được cập nhật kết quả từ tính toán của FB dựa trên các tham số vào.
- Dòng tín hiệu chạy từ trái qua phải.

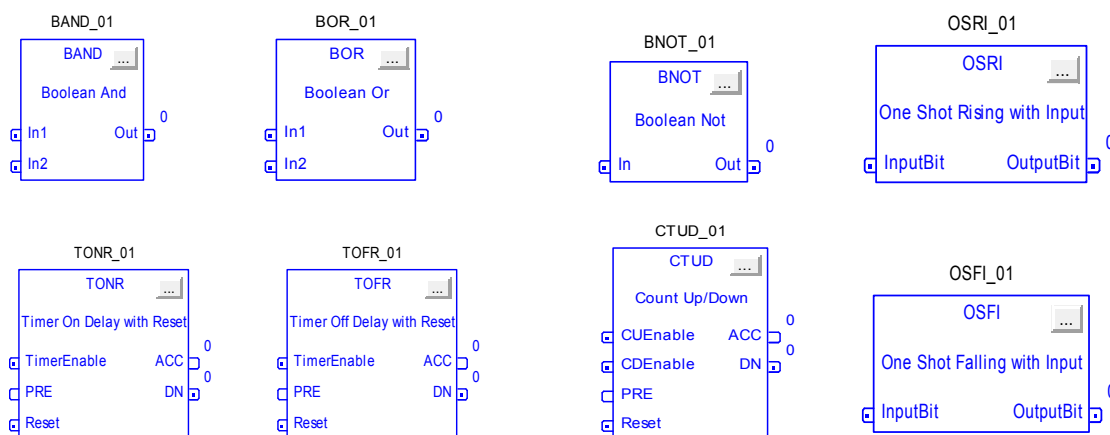


23

23

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ FBD



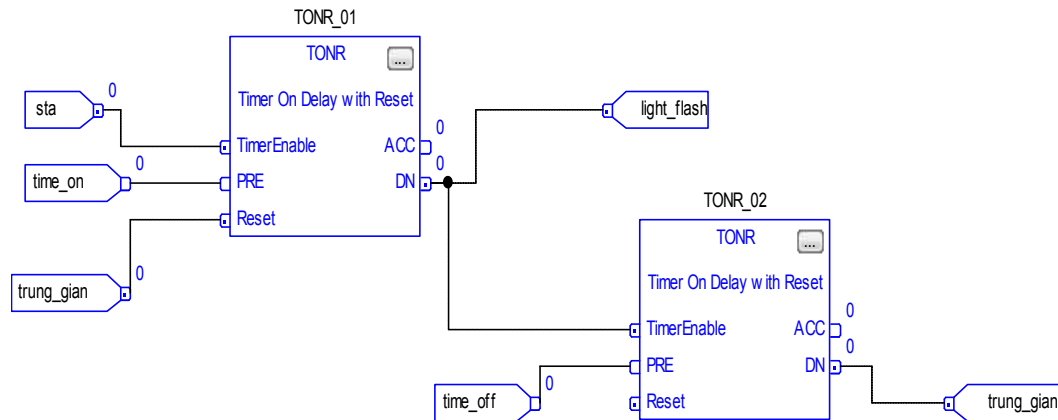
Một số khối chức năng trong FBD cho PLC CompactLogix

24

24

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ FBD



Chương trình sử dụng ngôn ngữ FBD cho PLC CompactLogix

25

25

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ FBD

- Ưu điểm:
  - ✓ Tương tự các mạch IC trong điện tử.
  - ✓ Ứng dụng kế hợp chức năng điều khiển logic và điều khiển phản hồi.
- Nhược điểm:
  - ✓ Hỗ trợ kém khi có một hay nhiều hành động lặp lại trong một khoảng thời gian định trước.
  - ✓ Khá cồng kềnh.

TDH-VD-BK

26

26

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ IL

- Ngôn ngữ lập trình bậc thấp, gồm các chuỗi câu lệnh, mỗi lệnh một dòng.
- Mỗi câu lệnh gồm một toán tử và một hay nhiều toán hạng.

Label	Operator	Operand	Comment
	LD	temp1	(*Load temp1 and*)
	GT	temp2	(*Test if temp1 > temp2*)
	JMPCN	Greater	(*Jump if not true to Greater*)
	LD	speed1	(*Load speed1*)
	ADD	200	(*Add constant 200*)
	JMP	End	(*Jump unconditional to End*)
Greater:	LD	speed2	(*Load speed2*)

27

27

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ IL

- Ưu điểm:
  - ✓ Thích hợp với ứng dụng nhỏ.
  - ✓ Tối ưu hóa bộ nhớ và tốc độ thực thi.
  - ✓ Có thể module hóa và tái sử dụng.
- Nhược điểm:
  - ✓ Ngôn ngữ bậc thấp, khó theo dõi.
  - ✓ Thanh ghi chỉ có một giá trị tại một thời điểm, khó làm việc với các dữ liệu có cấu trúc.

28

28



## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ IL

Toán tử	Ý nghĩa
LD	Đặt giá trị hiện tại cho toán hạng, nghịch đảo là LDN
ST	Đưa giá trị hiện tại tới địa chỉ toán hạng
S	Đặt toán hạng loại logic lên 1
R	Đặt lại logic 0 cho toán hạng
AND	Logic AND, nghịch đảo là ANDN
OR	Logic OR, nghịch đảo là ORN
XOR	Hoặc loại trừ
NOT	Logic nghịch đảo
ADD	Cộng
SUB	Trừ
MUL	Nhân

Toán tử	Ý nghĩa
DIV	Chia
MOD	Phép chia lấy dư
GT	So sánh lớn hơn
GE	So sánh lớn hơn hoặc bằng
EQ	So sánh bằng
NE	So sánh khác nhau
LE	So sánh nhỏ hơn hoặc bằng
LT	So sánh nhỏ hơn
JMP	Nhảy tới nhãn
CAL	Gọi khối chức năng
RET	Trở về từ gọi hàm, khối chức năng hay chương trình

29

29

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ ST

- Ngôn ngữ lập trình bậc cao (tương tự Pascal, C).
- 5 loại câu lệnh chính
  - ✓ Lệnh gán: *biến:=giá trị*
  - ✓ Lệnh lựa chọn: IF... THEN
  - ✓ Lệnh vòng lặp: WHILE, REPEAT, FOR
  - ✓ Function và function block
  - ✓ Lệnh điều khiển: RETURN, EXIT

TDH-VD-BK

30

30



## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ ST

- Lệnh điều kiện liệt kê CASE ... OF

*Case speed of:*

*Stop: rate: = 0.0; (\*Nếu speed bằng stop, gán rate bằng 0\*)*

*Slow: rate: = 20.4; (\*Nếu speed bằng slow, gán rate bằng 20.4\*)*

*Else*

*rate: = 0; (\*Các trường hợp còn lại rate bằng 0\*)*

*End\_case;*

TDH-VD-BK

33

33

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ ST

- Lệnh vòng lặp FOR

- Số vòng lặp là hữu hạn

- Cấu trúc:

*FOR <giá trị bắt đầu>*

*TO <giá trị kết thúc>*

*BY <bước nhảy> DO*

*<Các câu lệnh ...>*

*END\_FOR;*

Ví dụ:

*count := 0;*

*FOR i:=1 TO 10 DO*

*count := count + i;*

*END\_FOR;*

TDH-VD-BK

34

34

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ ST

- Lệnh vòng lặp WHILE
  - Số vòng lặp không xác định trước nhưng điều kiện kết thúc xác định
  - Cấu trúc:  
`WHILE <điều kiện> DO`  
     <Các câu lệnh>  
`END_WHILE;`

Ví dụ:

```
WHILE switch1 OR switch3
DO
    pump := FALSE;
    alarm := TRUE;
END_WHILE;
```

TDH-VD-BK

35

35

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ ST

- Lệnh vòng lặp REPEAT
  - Kiểm tra điều kiện sau khi thực hiện lệnh
  - Cấu trúc:  
`REPEAT`  
     <statements...>  
`UNTIL <boolean expression>`  
`END_REPEAT;`

Ví dụ:

```
B:=0
REPEAT
    B := B + 1;
UNTIL B > 10
END_REPEAT;
```

TDH-VD-BK

36

36

## 5.2. Các ngôn ngữ lập trình

### **Ngôn ngữ ST**

- Ưu điểm:
  - ✓ Phù hợp với tính toán phức tạp và vòng lặp
  - ✓ Dùng nhiều trong điều khiển tương tự
- Nhược điểm:
  - ✓ Đòi hỏi kiến thức nhất định về lập trình.

TDH-VD-BK

37

37

## 5.2. Các ngôn ngữ lập trình

### **Ngôn ngữ SFC**

- Phát triển từ Grafcet
- Có thể mạnh trong mô tả hệ thống điều khiển tuần tự.
- Sử dụng kết hợp với 4 ngôn ngữ còn lại

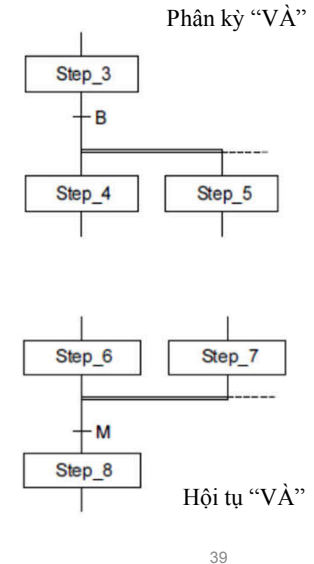
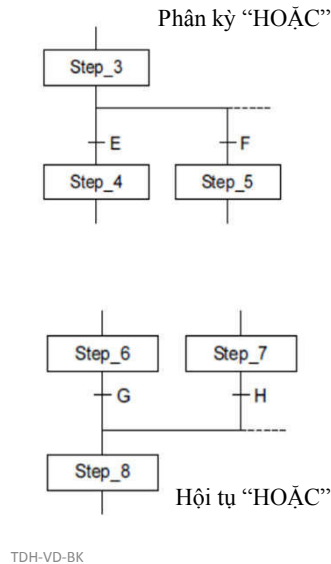
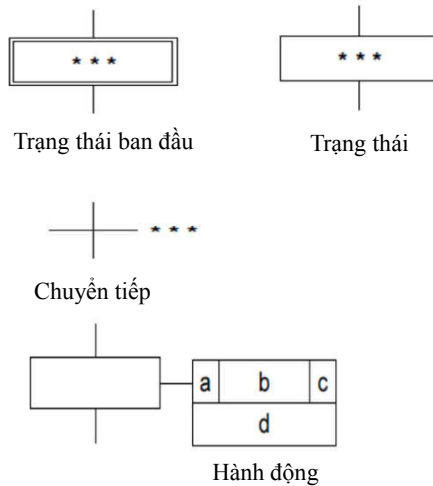
TDH-VD-BK

38

38

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ SFC



39

## 5.2. Các ngôn ngữ lập trình

### Ngôn ngữ SFC

- Ưu điểm:
  - ✓ Phù hợp với hệ thống tuần tự
- Nhược điểm:
  - ✓ Không phải ngôn ngữ hoàn chỉnh.
  - ✓ Thời gian thực hiện các phép điều kiện đơn giản lâu hơn các ngôn ngữ khác.

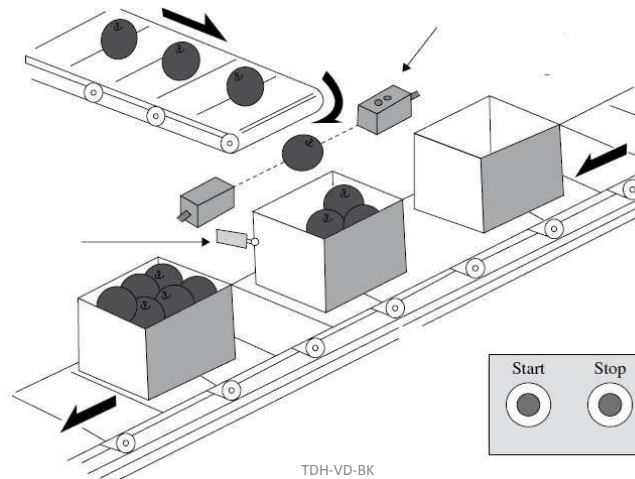
TDH-VD-BK

40

40

## 5.2. Các ngôn ngữ lập trình

### Ví dụ: Điều khiển dây chuyền đóng hộp sản phẩm



41

41

## 5.2. Các ngôn ngữ lập trình

### Mô tả công nghệ

- Khi nút Start được ấn, băng tải hộp chạy. Băng tải hộp sẽ dừng lại khi có tín hiệu từ cảm biến báo về rằng đã có hộp ở vị trí đóng gói. Lúc này, băng tải tảo bắt đầu hoạt động.
- Băng tải tảo làm nhiệm vụ cung cấp tảo đổ vào hộp. Một cảm biến được sử dụng để đếm từng quả tảo được đổ vào hộp.
- Khi số tảo đổ vào hộp bằng 10, băng tải tảo dừng, băng tải hộp lại chạy.
- Quá trình tiếp tục lặp lại cho đến khi nút Stop được ấn thì quy trình dừng ngay lập tức.

TDH-VD-BK

42

42

## 5.2. Các ngôn ngữ lập trình

### Các bước lập trình:

- Liệt kê tín hiệu vào/ra: 4 đầu vào số, 3 đầu ra số

STT	Tên tín hiệu	Loại tín hiệu	Đầu vào/Ra	
			Vào	Ra
1	Nút ấn khởi động - Start	Số	1	
2	Nút ấn dừng - Stop	Số	1	
3	Cảm biến hộp - CB_hop	Số	1	
4	Cảm biến tảo - CB_tao	Số	1	
5	Báo trạng thái làm việc (Lam_viec)	Số		1
6	Điều khiển bang tải hộp (Bang_hop)	Số		1
7	Điều khiển băng tải tảo (Bang_tao)	Số		1

43

43

## 5.2. Các ngôn ngữ lập trình

### Các bước lập trình:

- Lựa chọn PLC: CPU CompactLogix L32E, module IQ16 và OB16
- Phân địa chỉ vào/ra

STT	Tên tín hiệu	Loại tín hiệu	Đầu vào/Ra		Địa chỉ
			Vào	Ra	
1	Nút ấn khởi động - Start	Số	1		Local:2:I.Data.0
2	Nút ấn dừng - Stop	Số	1		Local:2:I.Data.1
3	Cảm biến hộp - CB_hop	Số	1		Local:2:I.Data.2
4	Cảm biến tảo - CB_tao	Số	1		Local:2:I.Data.3
5	Báo trạng thái làm việc (Lam_viec)	Số		1	Local:3:O.Data.0
6	Điều khiển bang tải hộp (Bang_hop)	Số		1	Local:3:O.Data.1
7	Điều khiển băng tải tảo (Bang_tao)	Số		1	Local:3:O.Data.2

TDH-VD-BK

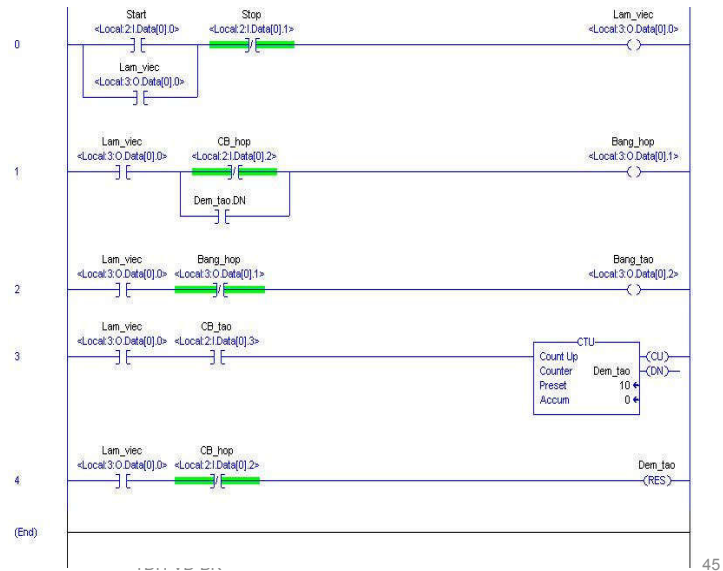
44

44



## 5.2. Các ngôn ngữ lập trình

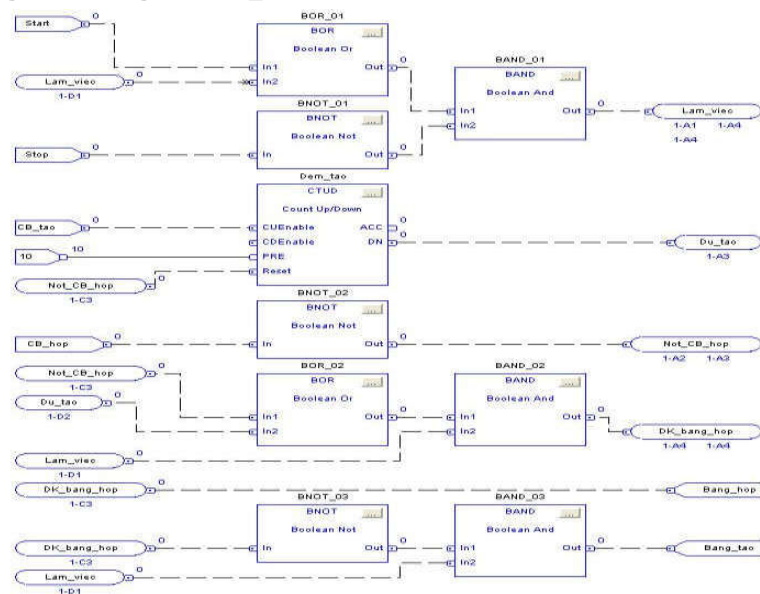
### Dùng LD



45

## 5.2. Các ngôn ngữ lập trình

### Dùng FBD



46

46

## 5.2. Các ngôn ngữ lập trình

### Dùng ST

```
Lam_viec [:=] (Start or Lam_viec) and not Stop;
CTUD(Dem_tao);
Dem_tao.CUEnable := CB_tao;
Dem_tao.pre := 10;
Dem_tao.Reset:= Lam_viec and not CB_hop;
Bang_hop [:=] Lam_viec and (( not CB_hop ) or
Dem_tao.DN);
Bang_tao [:=] Lam_viec and not Bang_hop;
```

TDH-VD-BK

47

47

## 5. Kỹ thuật lập trình PLC

5.1. Chu trình thiết kế chương trình PLC

5.2. Các ngôn ngữ lập trình

**5.3. Thiết kế chương trình sử dụng hàm logic**

5.4. Thiết kế chương trình sử dụng SFC

TDH-VD-BK

48

48

### 5.3. Thiết kế chương trình sử dụng hàm logic

Tìm hàm logic mô tả mối quan hệ giữa các tín hiệu vào ra

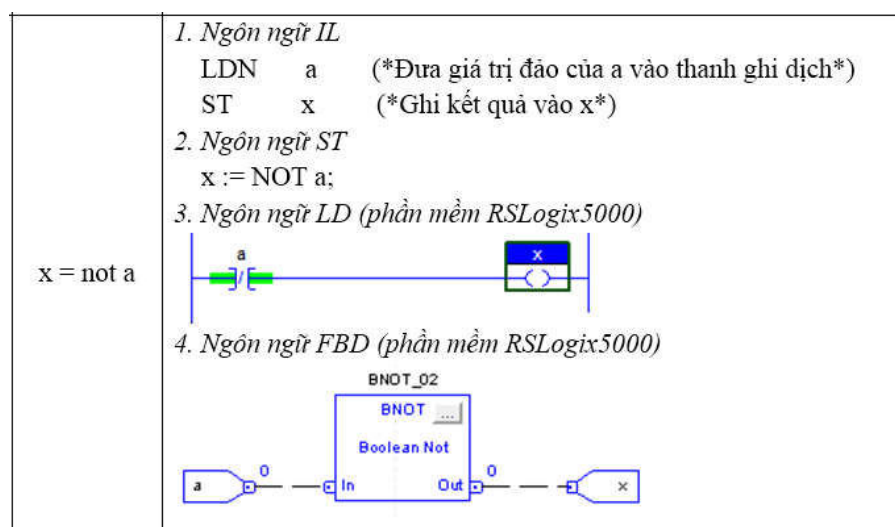
Chuyển hàm logic sang ngôn ngữ PLC tương ứng

TDH-VD-BK

49

49

### 5.3. Thiết kế chương trình sử dụng hàm logic

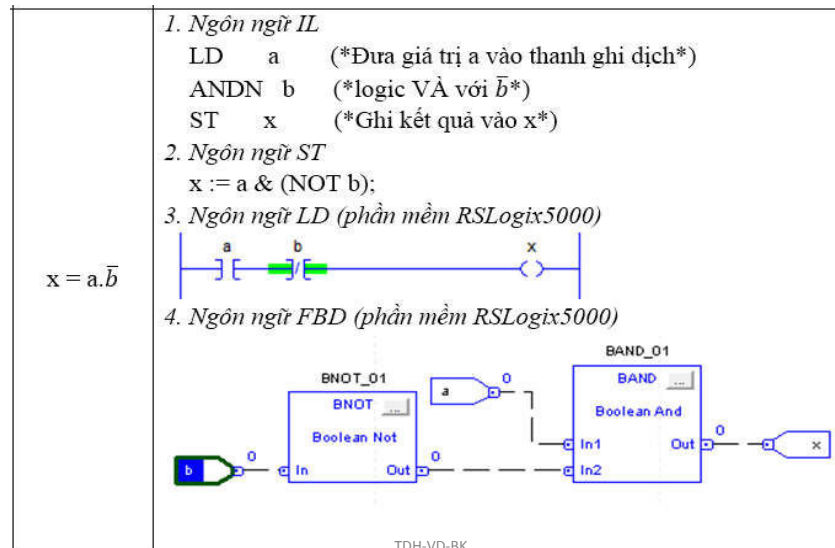


TDH-VD-BK

50

50

## 5.3. Thiết kế chương trình sử dụng hàm logic

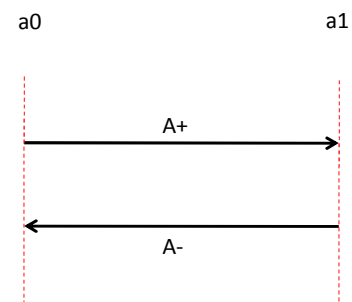


51

51

## 5.3. Thiết kế chương trình sử dụng hàm logic

- Ban đầu thiết bị chạm ở vị trí bên trái và tác động vào công tắc hành trình a0. Khi nút m được ấn, thiết bị di chuyển sang phải. Khi chạm vào a1 và thiết bị di chuyển về bên trái. Tiếp theo thiết bị lại chạm vào a0, nếu nút m được ấn thì chu trình được lặp lại.
- Chú ý rằng khi thiết bị rời khỏi vị trí của công tắc hành trình thì công tắc hành trình lại trở về trạng thái không tác động.



TDH-VD-BK

52

52

### 5.3. Thiết kế chương trình sử dụng hàm logic

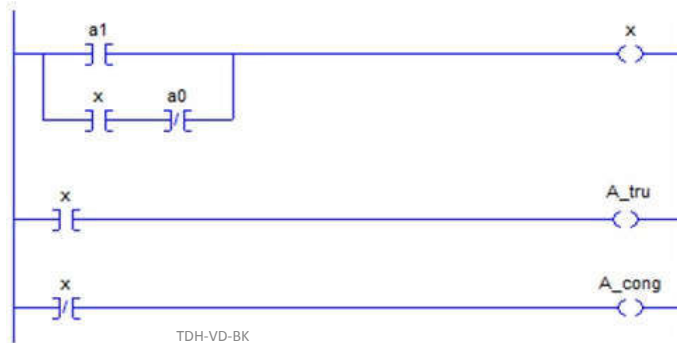
Sử dụng phương pháp ma trận trạng thái ta có thể tổng hợp các hàm logic mô tả công nghệ như sau:

$$X = a1 + \overline{a0}.X$$

$$A += \overline{X}$$

$$A -= X$$

LD:

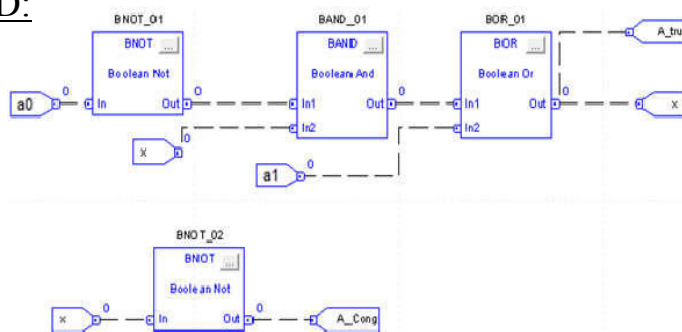


53

53

### 5.3. Thiết kế chương trình sử dụng hàm logic

FBD:



ST:

$X := a1 \text{ OR } (\text{NOT } a0 \text{ and } X);$

$A\_cong := \text{NOT } X;$

$A\_tru := X;$

TDH-VD-BK

54

IL:

**LDN a0**

**AND X**

**OR a1**

**ST X**

**LDN X**

**ST A\_cong**

**LDX**

**ST A\_tru**

54

## 5. Kỹ thuật lập trình PLC

5.1. Chu trình thiết kế chương trình PLC

5.2. Các ngôn ngữ lập trình

5.3. Thiết kế chương trình sử dụng hàm logic

**5.4. Thiết kế chương trình sử dụng SFC**

TDH-VD-BK

55

55

## 5.4. Thiết kế chương trình sử dụng SFC

**Lập Grafcet cho chu trình công nghệ**

**Chuyển Grafcet sang ngôn ngữ SFC**

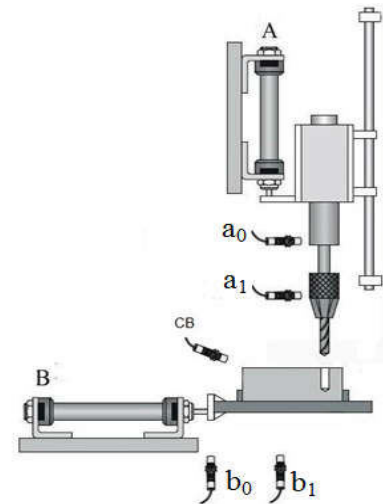
TDH-VD-BK

56

56

## 5.4. Thiết kế chương trình sử dụng SFC

- Xylanh A chuyển động lên-xuống (di chuyển mũi khoan) và xylanh B chuyển động phải-trái (đưa phôi vào vị trí cần khoan). Ban đầu các xylanh đều thu về.
- Khi có tín hiệu từ cảm biến CB báo hiệu có phôi, xylanh A sẽ thực hiện chuyển động đưa mũi khoan đi xuống A+ (xylanh B đứng im) và đến khi có tín hiệu từ cảm biến  $a_1$  thì sẽ thực hiện chuyển động thu về A- (xylanh B vẫn đứng im). Khi cảm biến  $a_0$  có tín hiệu thì xylanh A dừng, xylanh B thực hiện chuyển động sang phải (B+). Khi cảm biến  $b_1$  có tín hiệu thì xylanh B sẽ dừng và xylanh A lại thực hiện chuyển động đi xuống, khi gặp cảm biến  $a_1$  thì sẽ đi lên và khi gặp cảm biến  $a_0$  thì xylanh A dừng. Lúc này xylanh B thực hiện chuyển động sang trái (B-), đến khi cảm biến  $b_0$  có tín hiệu thì dừng. Sản phẩm được lấy ra bởi một cơ cấu khác và chu trình sẽ được lặp lại nếu tiếp tục có phôi.



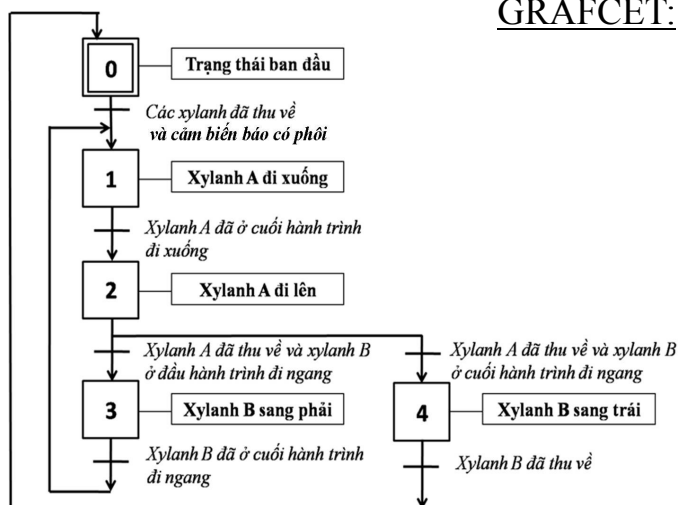
TDH-VD-BK

57

57

## 5.4. Thiết kế chương trình sử dụng SFC

GRAFCET:

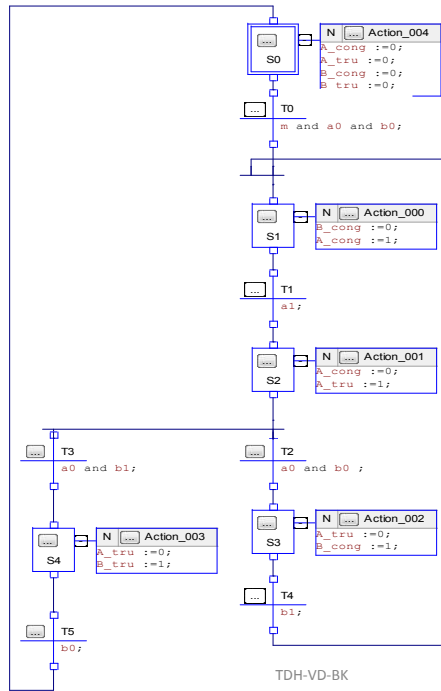


TDH-VD-BK

58

58

SFC:



59