

## CHƯƠNG IV : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

### I- KHÁI LƯỢC

#### 1- Công dụng

\* OB121 là khối tổ chức được gọi khi chương trình thực thi có lỗi. Nếu không có khối OB121, PLC sẽ chuyển sang trạng thái STOP.

\* CPU 312 không có OB 121.

\* Khi OB121 được sử dụng, mỗi khi hệ điều hành gọi, 20 byte biến tạm sẽ được hệ điều hành ghi.

Address	Dec	la	Name	Type	Initia	Comment
0.0	temp		OB121_EV_CLASS	BYTE		16#25, Event class 2, Entering event state, Internal fault event
1.0	temp		OB121_SW_FLT	BYTE		16#XX Software programming fault
2.0	temp		OB121_PRIORITY	BYTE		Priority of OB Execution
3.0	temp		OB121_OB_NUMBR	BYTE		121 (Organization block 121, OB121)
4.0	temp		OB121_BLK_TYPE	BYTE		16#88/8A/8B/8C/8E Type of block fault occurred in
5.0	temp		OB121_RESERVED_1	BYTE		Reserved for system
6.0	temp		OB121_FLT_REG	WORD		16#XX Specific register that caused fault
8.0	temp		OB121_BLK_NUM	WORD		Number of block that programming fault occurred in
10.0	temp		OB121_PRG_ADDR	WORD		Address in block where programming fault occurred
12.0	temp		OB121_DATE_TIME	DATE_A		Date and time OB121 started

Hình 4-1 : Các biến tạm của OB121

#### 2- Ý nghĩa giá trị các biến tạm :

a) Code lỗi OB121\_SW\_FLT (LB1):

OB121\_SW\_FLT là Code lỗi phần mềm của chương trình. Kết hợp với giá trị của OB121\_FLT\_REG(LW6), OB121\_RESERVED\_1(LB5) , cho ta biết về lỗi chương trình đã xảy ra. Nó lấy các giá trị:

-21H : Lỗi biến đổi BCD, giá trị không đúng dạng BCD.

Lúc này giá trị OB121\_FLT\_REG (LW6) chỉ số hiệu thanh ghi đang xảy ra lỗi ( 0H chỉ thanh ghi ACCU1).

-22H : Lỗi đọc ô nhớ ngoài vùng nhớ cho phép.

-23H : Lỗi ghi ô nhớ ngoài vùng cho phép

-28H : Lỗi đọc byte,word, double word với địa chỉ con trỏ sai ( giá trị bit khác “0” )

-29H : Lỗi ghi byte,word, double word với địa chỉ con trỏ sai ( giá trị bit khác “0” )

Lúc này giá trị OB121\_FLT\_REG (LW6) chỉ địa chỉ byte bị sai.

Và OB121\_RESERVED\_1 : Chỉ cụ thể lỗi :

## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

Giá trị các bit 7-4 chỉ kiểu dữ liệu truy cập :

0: truy cập bit

1: truy cập byte

2 : truy cập word

3: truy cập double word

Giá trị các bit 3-0 chỉ vùng nhớ :

0 : Vùng nhớ I/O (PI/PO)

1: Vùng nhớ I

2: Vùng nhớ Q

3: Vùng nhớ M

4: Vùng DB

5: Vùng DI

6: Vùng nhớ L của khối

7: Vùng nhớ L của khối gọi

- 24H : Lỗi đọc ô nhớ có địa chỉ sai cấu trúc

- 25H : Lỗi ghi ô nhớ có địa chỉ sai cấu trúc

Lúc này OB121\_FLT\_REG , byte thấp, chỉ thị ID của vùng lỗi (Khi lỗi trong vùng L của khối, nó có giá trị B#16#86) .

- 26H : Lỗi số hiệu bộ định thời sai

- 27H : Lỗi số hiệu bộ đếm sai

Lúc này OB121\_FLT\_REG chỉ thị số bị lỗi.

- 30H : Lỗi ghi vào DB chỉ cho phép đọc

- 31H : Lỗi ghi vào DI chỉ cho phép đọc

- 32H : Lỗi sai số hiệu DB khi truy cập ( Ngoài vùng cho phép)

- 33H : Lỗi sai số hiệu DI khi truy cập

Lúc này OB121\_FLT\_REG chỉ thị số hiệu DB, DI sai.

- 34H : Lỗi gọi sai số hiệu FC

- 35H : Lỗi gọi sai số hiệu FB

- 3AH : Lỗi truy cập DB không được nạp vào PLC, số hiệu DB trong vùng cho phép.

- 3CH : Lỗi gọi FC không được nạp, số hiệu FC trong vùng cho phép.

- 3EH : Lỗi gọi FB không được nạp, số hiệu FB trong vùng cho phép.

Lúc này OB121\_FLT\_REG chỉ thị số hiệu bị lỗi.

b) Kiểu khối logic chương trình bị lỗi OB121\_BLK\_TYPE (LB4)

Byte này chứa thông tin kiểu khối, không sử dụng trong S7-300. Khi sử dụng PLCSIM, vì phần mềm dùng chung cho cả S7-400 nên thông tin này được hệ điều hành ghi vào OB121.

- B#16#88 : OB

- B#16#8A : DB

- B#16#8C : FC

- B#16#8E : FB

## II-CÁC VÍ DỤ ỨNG DỤNG

### 1- Theo dõi hoạt động của khối OB121

\* Với PLC thực, ta có thể theo dõi khi kết nối với PC và quan sát các giá trị trong bảng biến, hoặc dùng các đầu ra và thể hiện các lỗi nhớ chương trình trong OB121.

\* Trong khối OB121 ta viết chương trình chuyển các giá trị biến tạm từ LB0 đến LB11 vào các ô nhớ MB100 đến MB111.

\* Ta có thể gán một số biến ra đầu ra để theo dõi một số lỗi xác định trước.

\* Cần lưu ý là khi đã lập trình và nạp vào PLC khối OB121, CPU sẽ không chuyển sang trạng thái STOP khi có lỗi lập trình, chỉ có đèn SF sáng báo có lỗi.

### 2- Ví dụ với lỗi biến đổi BCD mà giá trị không đúng kiểu BCD

\* Khi sử dụng lệnh biến đổi BTI, thanh ghi ACCU1 cần đang chứa dữ liệu dạng BCD, tức mỗi nhóm 4 bit chỉ chứa các giá trị từ 0000 đến 1001. Nếu nó chứa dữ liệu lớn hơn (1010 đến 1111) thì lệnh không thể thực thi và chương trình bị lỗi. Nếu không có OB121, PLC sẽ chuyển sang trạng thái STOP.

\* Giá trị OB121\_SW\_FLT :

B#16#21

\* Chương trình ví dụ :

Trong OB1 ta viết đoạn lệnh gọi FC1:

Call FC1

Trong FC1 :

A I 0.0

CU C 1

A I 0.1

R C 1

L C 1 //Giá trị nhận được dạng INT

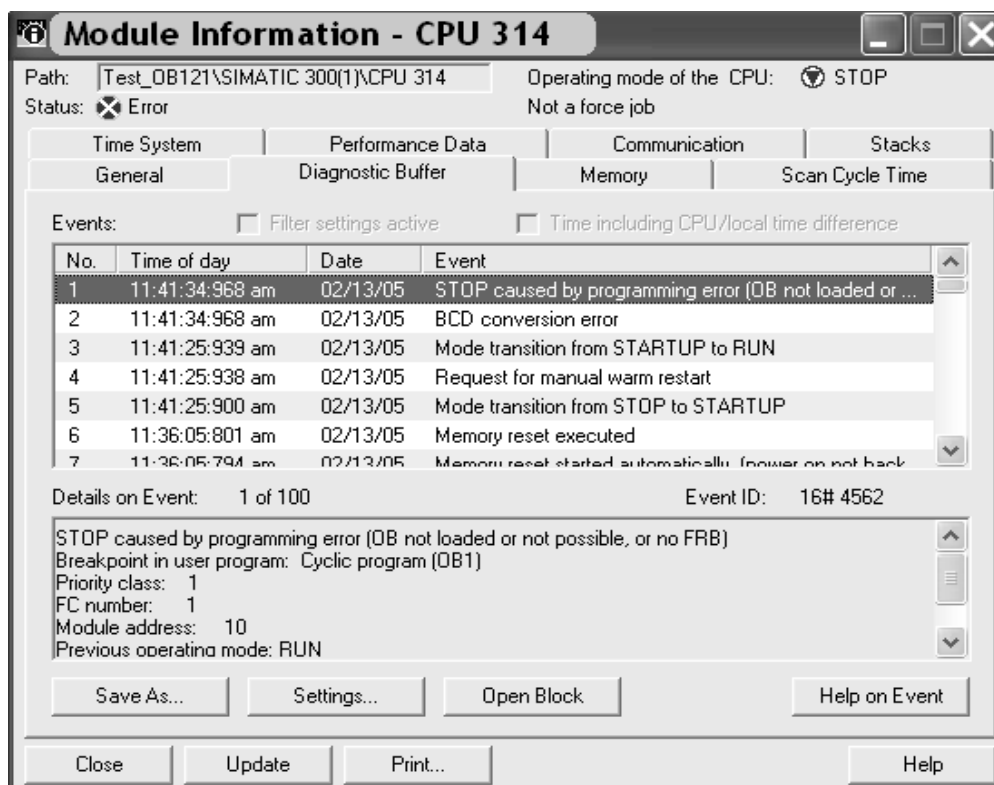
BTI

T QB 5

\* Chương trình bị lỗi do dùng lệnh BTI với số INT. Khi giá trị C1 nhỏ hơn 10, PLC không bị lỗi vì giá trị INT hay BCD viết giống nhau. Khi C1 có giá trị từ 10 trở lên, cách viết khác nhau và PLC phát hiện lỗi, treo CPU.

\* Đọc trong **Module Information** có thể phát hiện lỗi này:

## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85



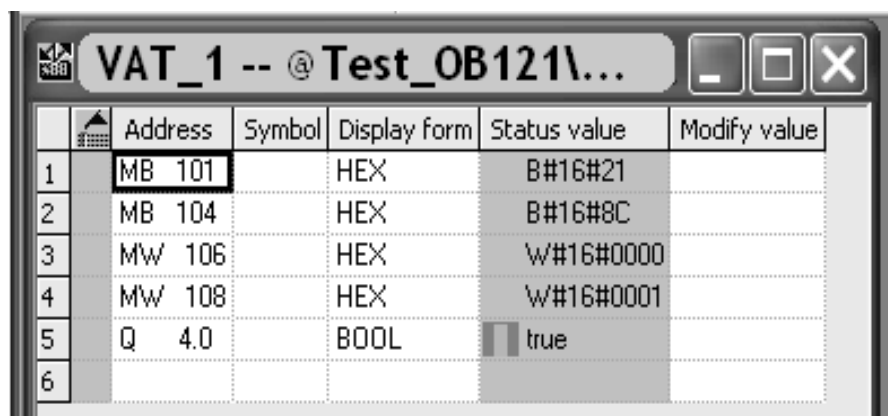
Hình 4-2: Xác định lỗi biến đổi BCD bằng Module Information

\*Nếu ta sử dụng khối OB121, PLC sẽ không chuyển sang trạng thái STOP, đèn SF sáng. Chương trình sau trong OB121 cho phép ta theo dõi hoạt động thông báo lỗi.

```
CALL "BLKMOV"
SRCBLK :=P#L 0.0 BYTE 20
RET_VAL:=MW200
DSTBLK :=P#M 100.0 BYTE 20
SET
= Q 4.0
L LB1
L B#16#21
==I
JCN N001
SET
= Q 4.1
N001: NOP 0
```

\* Khi giá trị bộ đếm đạt đến 10, chương trình bị lỗi, OB121 được gọi và đầu ra Q4.0 lấy giá trị 1.

\* Do lỗi là sai kiểu dữ liệu BCD, Code lỗi là B#16#21 nên Q4.1 lên mức 1.



	Address	Symbol	Display form	Status value	Modify value
1	MB 101		HEX	B#16#21	
2	MB 104		HEX	B#16#8C	
3	MW 106		HEX	W#16#0000	
4	MW 108		HEX	W#16#0001	
5	Q 4.0		BOOL	true	
6					

Hình 4-3 : Bảng biến để theo dõi lỗi

Theo kết quả trên bảng biến, ta biết chương trình bị lỗi chuyển dữ liệu kiểu BCD ( MB101 = B#16#21) ; Kiểu khối chứa lệnh phạm lỗi là FC (MB104 = B#16#8C”) và số hiệu khối là 1 (MW108= W#16#0001 ), Vị trí của lỗi trên thanh ghi ACCU1 ( MW106 = W#16#0000 ). Bảng biến này được sử dụng với PLCSIM nên MB104 có thông tin.

### 3- Lỗi khi sử dụng ngoài vùng nhớ cho phép.

a) Giới hạn vùng nhớ:

\*Tùy thuộc loại CPU sử dụng, độ rộng vùng nhớ bị giới hạn. Ta có thể xác định giới hạn này theo đặc tính kỹ thuật của CPU. Nói chung các CPU đời sau có độ rộng vùng nhớ lớn hơn CPU đời trước. Riêng PLCSIM có vùng nhớ rất rộng.

Ví dụ :

	CPU 312IFM	CPU 314	PLCSIM
Bit Memory	1024 (MB0 - MB127)	2048 (MB0 – MB 255)	131.072 (MB0 – MB16.368)
Counter	C0 – C31	C0 – C64	C0 – C511
Timer	T0 – T63	T0 – T127	T0 – T511
Data Blocks	DB1 – DB63	DB1-DB127	DB1-DB4095
FC	FC0 – FC 31	FC0 – FC127	FC0 – FC2047
FB	FB0 – FB31	FB0 – FB127	FB0 - FB2047

\* Lỗi thường xảy ra khi sử dụng định địa chỉ gián tiếp. Nếu sử dụng định địa chỉ trực tiếp, lỗi sẽ được phát hiện ngay khi nạp chương trình cho PLC (Chương trình báo lỗi, không thể nạp được)

b) Lỗi đọc ghi ô nhớ ngoài vùng .

\* Lỗi đọc, ghi ô nhớ ngoài vùng cần thực hiện với PLC thực. Với PLCSIM, vùng nhớ cho phép rất lớn (131.072 bit, 16,384Kbyte), chương trình không bị lỗi dù nó không hoạt động được trên PLC thực.

\* Chương trình ví dụ lỗi ghi ô nhớ ngoài vùng cho phép:

## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

+ Trong OB1 viết chương trình:

```
CALL FC 1  
L 5.000000e+000  
T MD [MD 0]
```

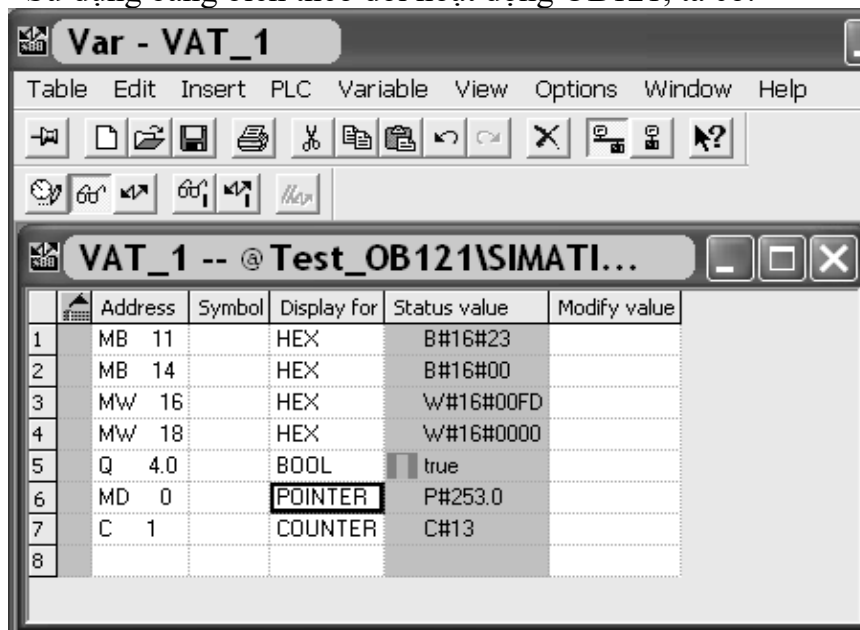
+ Trong FC1 viết chương trình:

```
A I 0.0  
CU C 1  
A I 0.1  
R C 1  
L C 1  
L 240 // Thử từ byte 240  
+I  
SLD 3  
T MD 0
```

+ Trong OB121 viết chương trình :

```
CALL "BLKMOV"  
SRCBLK :=P#L 0.0 BYTE 20  
RET_VAL:=MW4  
DSTBLK :=P#M 10.0 BYTE 20  
SET  
= Q 4.0
```

\*Sử dụng bảng biến theo dõi hoạt động OB121, ta có:



	Address	Symbol	Display for	Status value	Modify value
1	MB 11		HEX	B#16#23	
2	MB 14		HEX	B#16#00	
3	MW 16		HEX	W#16#00FD	
4	MW 18		HEX	W#16#0000	
5	Q 4.0		BOOL	true	
6	MD 0		POINTER	P#253.0	
7	C 1		COUNTER	C#13	
8					

Hình 4-4: Lỗi ghi ô nhớ ngoài vùng cho phép

\* Nếu chương trình đọc ô nhớ, Code lỗi sẽ có giá trị B#16#22.

\* Ta cũng có thể phát hiện lỗi nhờ **Module Information**.

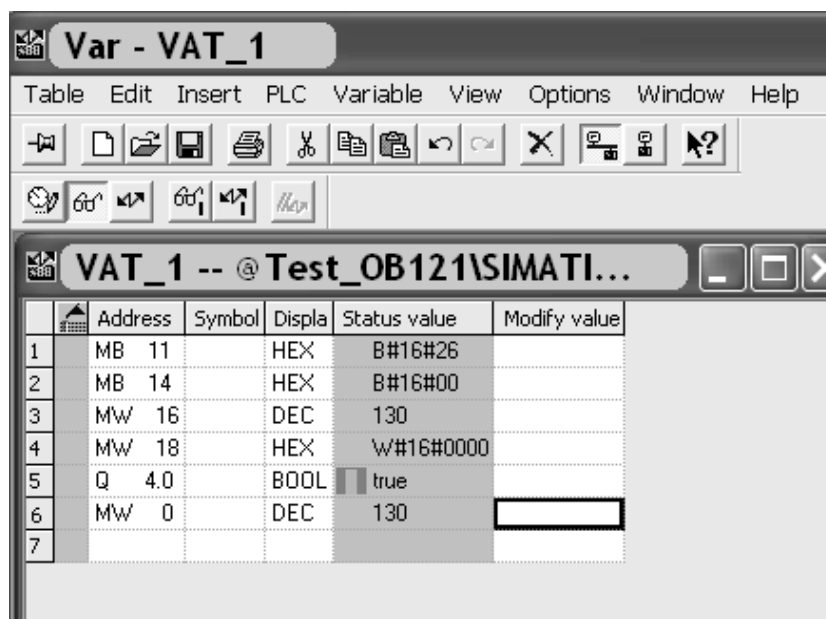
c) Lỗi sai số hiệu bộ định thời, bộ đếm, khối dữ liệu.

\* Nếu số hiệu bộ đếm, định thời, khối dữ liệu (DB) mà nằm ngoài vùng giới hạn ,PLC bị lỗi.

\* Ví dụ :

```

A   I   0.0
CU   C   1
A   I   0.1
R   C   1
L   C   1
L   120
+I
T   MW   0
AN   T [MW 0]
L   S5T#2S
SE   T [MW 0]
    
```



Hình 4-3: Lỗi sai số hiệu bộ định thời

\* Giá trị Code lỗi cho biết đây là lỗi sai số hiệu bộ định thời (B#16#26). Nếu lỗi sai số hiệu bộ đếm, giá trị này là B#16#27. Nếu sai số hiệu DB, giá trị này là B#16#32.

#### 4- Lỗi sai con trỏ địa chỉ.

a) Khi sử dụng địa chỉ gián tiếp, địa chỉ của ô nhớ được chỉ bằng dữ liệu kiểu con trỏ. Dữ liệu kiểu con trỏ được tạo ra từ các nguồn khác nhau. Có 2 điều kiện phải thỏa mãn :

\* Nếu truy cập byte, word, Double Word thì 3 bit nhỏ ( chỉ địa chỉ bit) phải bằng 0, nếu không chương trình sẽ bị lỗi.

\* Nếu truy cập thanh ghi kiểu chéo vùng, trong thanh ghi địa chỉ, bit thứ 31 phải có giá trị 1. Nếu không chương trình sẽ bị lỗi.

\* Code lỗi lấy giá trị :

+ B#16#28, B#16#29 : Địa chỉ bit khác 0

## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

+ B#16#24,B#16#25 : Sai cấu trúc bit 31

b) Ví dụ 1 :

\* Trong OB1 viết :

Call FC1

\* Trong FC1 :

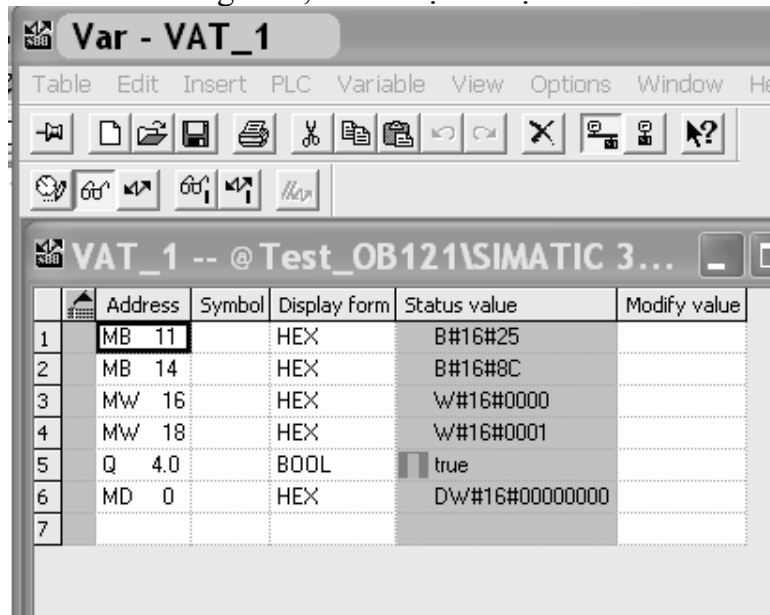
LAR1 P#4.0

T MD 0

L L#5

T D [AR1,P#0.0]

\* Theo dõi bảng biến, ta xác định được lỗi



	Address	Symbol	Display form	Status value	Modify value
1	MB 11		HEX	B#16#25	
2	MB 14		HEX	B#16#8C	
3	MW 16		HEX	W#16#0000	
4	MW 18		HEX	W#16#0001	
5	Q 4.0		BOOL	true	
6	MD 0		HEX	DW#16#00000000	
7					

Hình 4-5 : Lỗi sai cấu trúc con trỏ địa chỉ

c) Ví dụ 2 :

\* Trong FC0 ta viết:

L P#M 50.0

T MD 14

A I 0.0

CU C 1

A I 0.1

R C 1

L C 1

L 3

/I

ITD

L MD 14

+D

LAR1

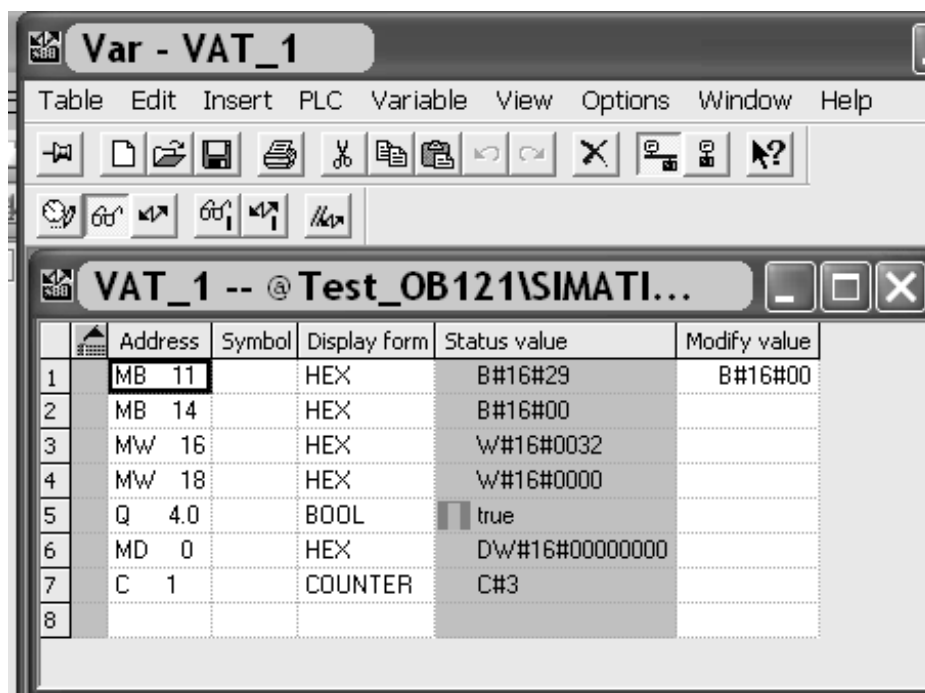
L 5.000000e+000

T D [AR1,P#0.0]



## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

\* Chương trình sẽ bị lỗi khi  $C1 \geq 3$  do kết quả phép chia khác 0, giá trị bit thanh ghi địa chỉ khác 0 khi ta truy cập DW.



	Address	Symbol	Display form	Status value	Modify value
1	MB 11		HEX	B#16#29	B#16#00
2	MB 14		HEX	B#16#00	
3	MW 16		HEX	W#16#0032	
4	MW 18		HEX	W#16#0000	
5	Q 4.0		BOOL	true	
6	MD 0		HEX	DW#16#00000000	
7	C 1		COUNTER	C#3	
8					

Hình 4-6 : Lỗi giá trị bit của con trỏ khác 0 khi ghi DW

### 5- Lỗi gọi các khối logic không được nạp.

a) Khi sử dụng chương trình có cấu trúc, ta phải sử dụng các lệnh gọi các khối logic FC, FB, DB. Khi viết chương trình, nếu các khối này không tồn tại, chương trình sẽ báo lỗi (Khi sử dụng kiểm lỗi ngay từng dòng lệnh).

\* Nếu việc nạp chương trình cho PLC không đầy đủ, có khối logic bị thiếu, PLC sẽ bị lỗi khi có lệnh gọi nó.

\* Code lỗi có giá trị :

-B#16#3A : Truy cập DB không được nạp.

-B#16#3C : Gọi FC không được nạp.

-B#16#3E : Gọi FB không được nạp

b) Ví dụ :

\* Trong FC2 ta gọi FC3, nhưng FC3 không được nạp vào PLC

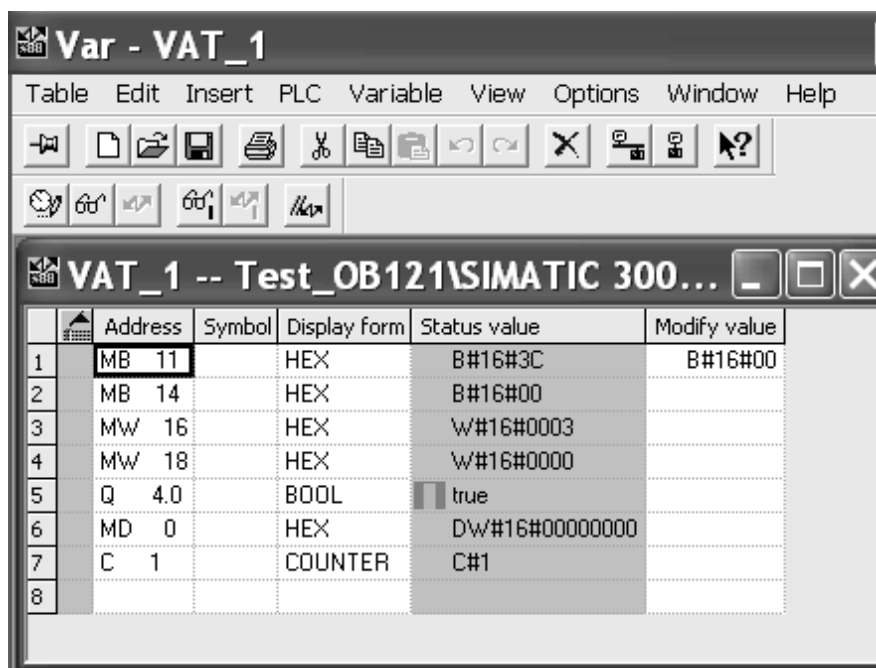
\* Khi hoạt động PLC bị lỗi . Nếu không có OB121, PLC sẽ dừng.

\* Nếu có OB121, nó sẽ được gọi và PLC bỏ qua lỗi, tiếp tục hoạt động.

\* Ta có thể theo dõi lỗi nhờ bảng biến, hoặc các đầu ra báo hiệu.

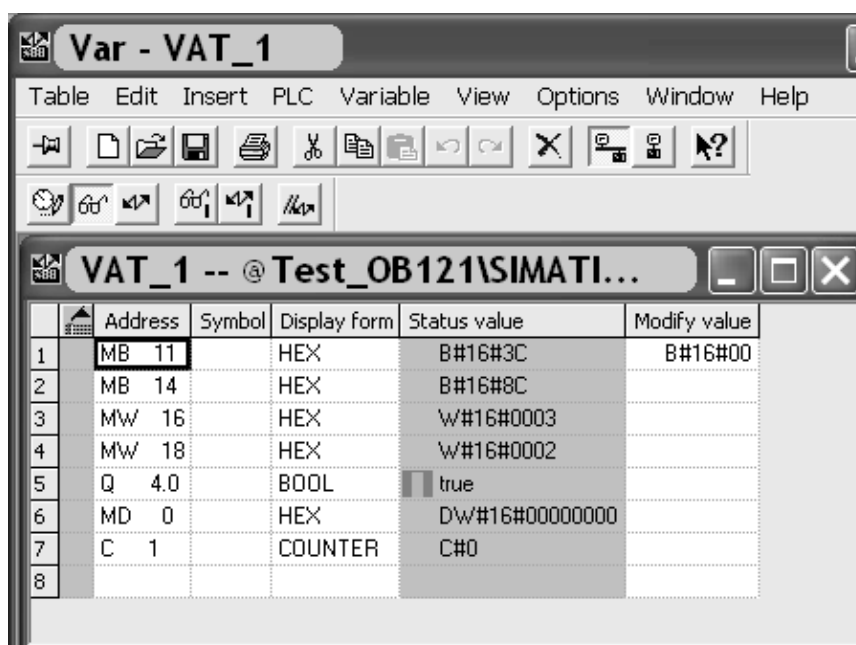
\* Ta đọc được khối logic không được nạp là FC3.

\* Khi sử dụng PLCSIM, ta đọc được thêm khối phát lệnh gọi là FC2.



	Address	Symbol	Display form	Status value	Modify value
1	MB 11		HEX	B#16#3C	B#16#00
2	MB 14		HEX	B#16#00	
3	MW 16		HEX	W#16#0003	
4	MW 18		HEX	W#16#0000	
5	Q 4.0		BOOL	true	
6	MD 0		HEX	DW#16#00000000	
7	C 1		COUNTER	C#1	
8					

Hình 4-7 : Lỗi FC không được nạp, PLC thực

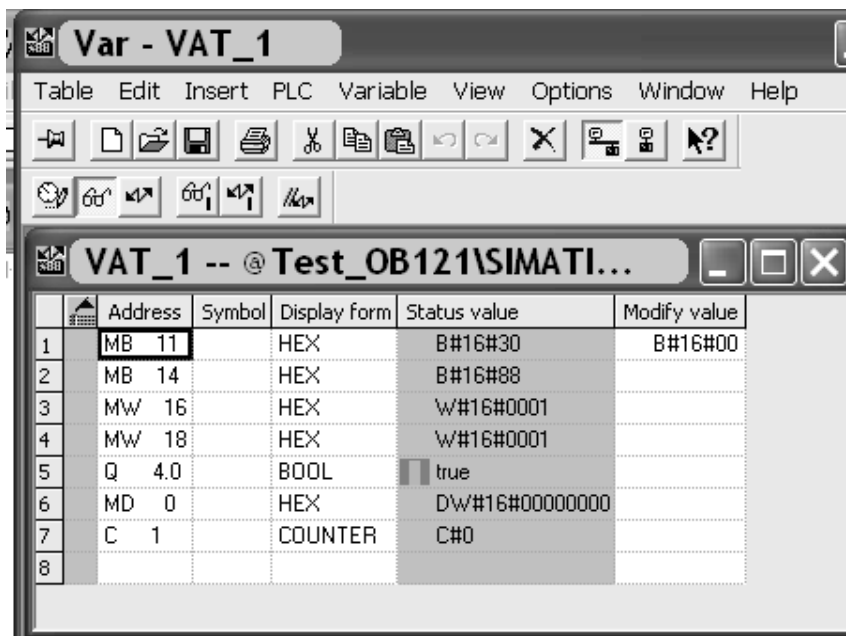


	Address	Symbol	Display form	Status value	Modify value
1	MB 11		HEX	B#16#3C	B#16#00
2	MB 14		HEX	B#16#8C	
3	MW 16		HEX	W#16#0003	
4	MW 18		HEX	W#16#0002	
5	Q 4.0		BOOL	true	
6	MD 0		HEX	DW#16#00000000	
7	C 1		COUNTER	C#0	
8					

Hình 4-7 : Lỗi FC gọi không được nạp, PLCSIM

## 6- Lỗi ghi vào DB chỉ cho phép đọc

a) Khi tạo các khối dữ liệu DB,DI, ta có thể đặt bảo vệ cấm ghi, nhằm bảo vệ giá trị dữ liệu trong quá trình hoạt động. Khi khối dữ liệu bị ghi đè, nó sẽ lưu giá trị mới cho tới khi được ghi bằng giá trị khác hoặc khối dữ liệu đó được nạp lại từ chương trình.



Hình 4-8 : Lỗi ghi vào DB chỉ được phép đọc

### III- SỬ DỤNG OB85

#### 1- Công dụng

\* Khi lập trình sử dụng các ngắt định thời, ta sử dụng các hàm hệ thống để xác định thời điểm ngắt và tích cực ngắt ( SFC28, SFC30 với OB10, SFC 32 với OB20).

\* Nếu ngắt định thời đã đặt và tích cực, nếu không có OB tương ứng được nạp vào PLC, chương trình bị lỗi và chuyển sang STOP.

\* Nếu sử dụng OB85, PLC sẽ không chuyển sang STOP. Lúc đó , ta có thể biết khối OB bị thiếu.

#### 2- Ví dụ lập trình

\* Lập trình phát hiện khối OB bị thiếu:

+ Trong OB1 viết chương trình đặt điểm gọi OB10 và OB20

```
CALL "D_TOD_DT"
IN1  :=MW2
IN2  :=MD4
RET_VAL:=#CALL_ob10_dt
CALL "SET_TINT"
OB_NR :=10
SDT   :=#CALL_ob10_dt
PERIOD :=W#16#201
RET_VAL:=MW200
CALL "ACT_TINT"
```

## CHƯƠNG 4 : XỬ LÝ LỖI CHƯƠNG TRÌNH SỬ DỤNG OB121, OB85

```
OB_NR :=10
RET_VAL:=MW202
CALL "DT_DATE"
IN  :=#OB1_DATE_TIME
RET_VAL:=MW8
CALL "DT_TOD"
IN  :=#OB1_DATE_TIME
RET_VAL:=MD10
```

```
A  I  0.7
=  Q  8.0
A  I  0.0
FP  M  0.0
JCN n01
CALL "SRT_DINT"
OB_NR :=20
DTIME :=T#4S
SIGN  :=W#16#1
RET_VAL:=MW220
```

n01: NOP 0

+ Trong OB85 ta viết chương trình báo loại OB thiếu :

```
L  #OB85_OB_NUM
L  10
==I
JCN n01
SET
=  Q  4.0
```

n01: L #OB85\_OB\_NUM

```
L  20
==I
JCN n02
SET
=  Q  4.1
```

n02: NOP 0

\* Khi cho chương trình thực thi , PLC không chuyển sang STOP, và đầu ra Q4.0 lên mức 1 khi thiếu OB10, còn đầu ra Q4.1 lên mức 1 khi thiếu OB20.