

Nhóm 1 - CE118.Q11.VMTN

# THIẾT KẾ VI XỬ LÝ 16-BIT

theo kiến trúc tập lệnh đơn giản

GV hướng dẫn: Tạ Trí Đức

# DANH SÁCH THÀNH VIÊN

- 24520344 Nguyễn Việt Dũng (Nhóm trưởng)
- 24520301 Tạ Vương Điền
- 24521177 Huỳnh Hữu Nguyên
- 24520839 Phan Công Đăng Khoa
- 24520816 Lê Đăng Khoa

→  
-MỤC LỤC-

01

TỔNG QUAN  
THIẾT KẾ

02

PIPELINE &  
MẠCH NHÂN

03

KIỂM ĐỊNH

# TẬP LỆNH

- RRR:

Ins	Opcode	Rs	Rt	Rd	Funct	Assembly-Code Format	Meaning
	15 13	12 10	9 7	6 4	3 0		
add	000	Rs	Rt	Rd	0000	add Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} + \text{Reg[Rt]}$
inc	000	Rs	000	Rd	0001	inc Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} - \text{Reg[Rt]}$
sub	000	Rs	Rt	Rd	0010	sub Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} + 1$
dec	000	Rs	000	Rd	0011	dec Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} - 1$
and	000	Rs	Rt	Rd	0100	and Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \& \text{Reg[Rt]}$
or	000	Rs	Rt	Rd	0101	or Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]}   \text{Reg[Rt]}$
nand	000	Rs	Rt	Rd	0110	nand Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \sim\& \text{Reg[Rt]}$
xor	000	Rs	Rt	Rd	0111	xor Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \wedge \text{Reg[Rt]}$
shfl	000	Rs	000	Rd	1000	shfl Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \ll 1$
shfr	000	Rs	000	Rd	1001	shfr Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 1$
sll	000	Rs	000	Rd	1010	sll Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \ll 2$
srr	000	Rs	000	Rd	1011	shrr Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 2$
sra	000	Rs	000	Rd	1100	shra Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 1$ (giữ bit dấu)
mul	000	Rs	Rt	Rd	1101	mul Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \times \text{Reg[Rt]}$
slt	000	Rs	Rt	Rd	1110	slt Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow (\text{Reg[Rs]} < \text{Reg[Rt]}) ? 1 : 0$
sgt	000	Rs	Rt	Rd	1111	sgt Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow (\text{Reg[Rs]} > \text{Reg[Rt]}) ? 1 : 0$

- RI:

Ins	Opcode	Rs	Imm	Assembly-Code Format	Meaning
	15 13	12 10	9 0		
lw	100	Rs	Imm	lw Rs, Imm	$\text{Reg[Rs]} \leftarrow \text{Mem[Imm]}$
sw	101	Rs	Imm	sw Rs, Imm	$\text{Mem[Imm]} \leftarrow \text{Reg[Rs]}$
jnz	110	Rs	Imm	jnz Rs, Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} \neq 0) ? \text{Imm} : \text{PC} + 1$
li	111	Rs	Imm	li Rs, Imm	$\text{Reg[Rs]} \leftarrow \text{Imm}$

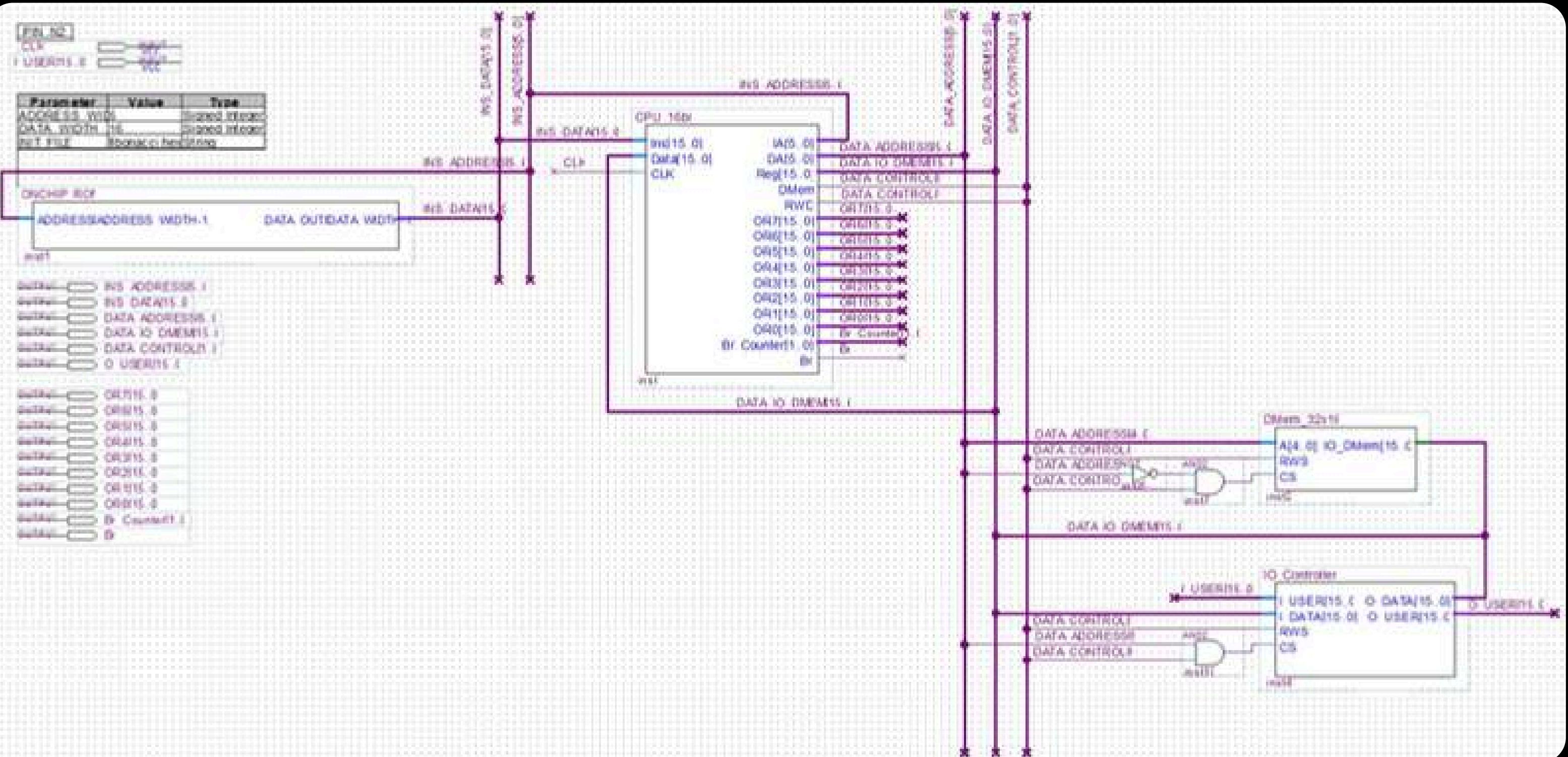
- Extend (Mã giá):

Ins	Opcode	Rs	Rt	Rd	Funct	Assembly-Code Format	Meaning
	15 13	12 10	9 7	6 4	3 0		
not	000	Rs	Rs	Rd	1000	not Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \sim\& \text{Reg[Rs]}$
mov	000	Rs	Rs	Rd	0110	mov Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \& \text{Reg[Rs]}$
nop	001	Rs	Rs		0000000	nop	$\text{Reg[Rs]} \leftarrow \text{Reg[Rs]} + 0$
jmp	011	Rs	Rs		Imm	jmp Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} == \text{Reg[Rs]}) ? \text{Imm} : \text{PC} + 1$

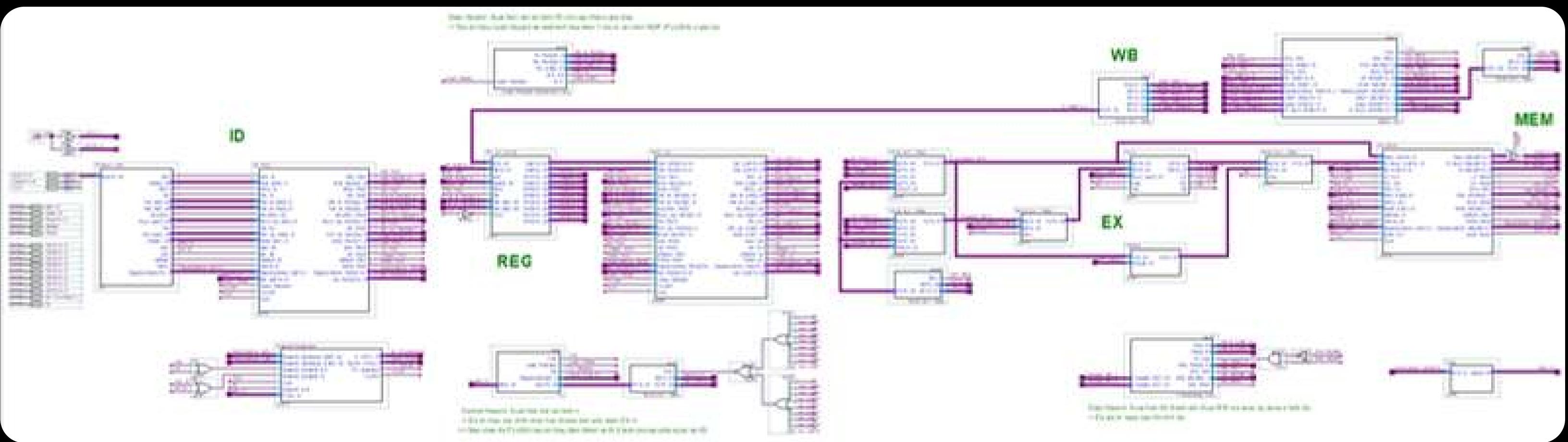
- RRI:

Ins	Opcode	Rs	Rt	Imm	Assembly-Code Format	Meaning
	15 13	12 10	9 7	6 0		
addi	001	Rs	Rt	Imm	addi Rs, Rt, Imm	$\text{Reg[Rt]} \leftarrow \text{Reg[Rs]} + \text{SignExtImm}$
subi	010	Rs	Rt	Imm	subi Rs, Rt, Imm	$\text{Reg[Rt]} \leftarrow \text{Reg[Rs]} - \text{SignExtImm}$
beq	011	Rs	Rt	Imm	beq Rs, Rt, Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} == \text{Reg[Rt]}) ? \text{Imm} : \text{PC} + 1$

# DATAPATH



# DATAPATH - CPU 16-BIT



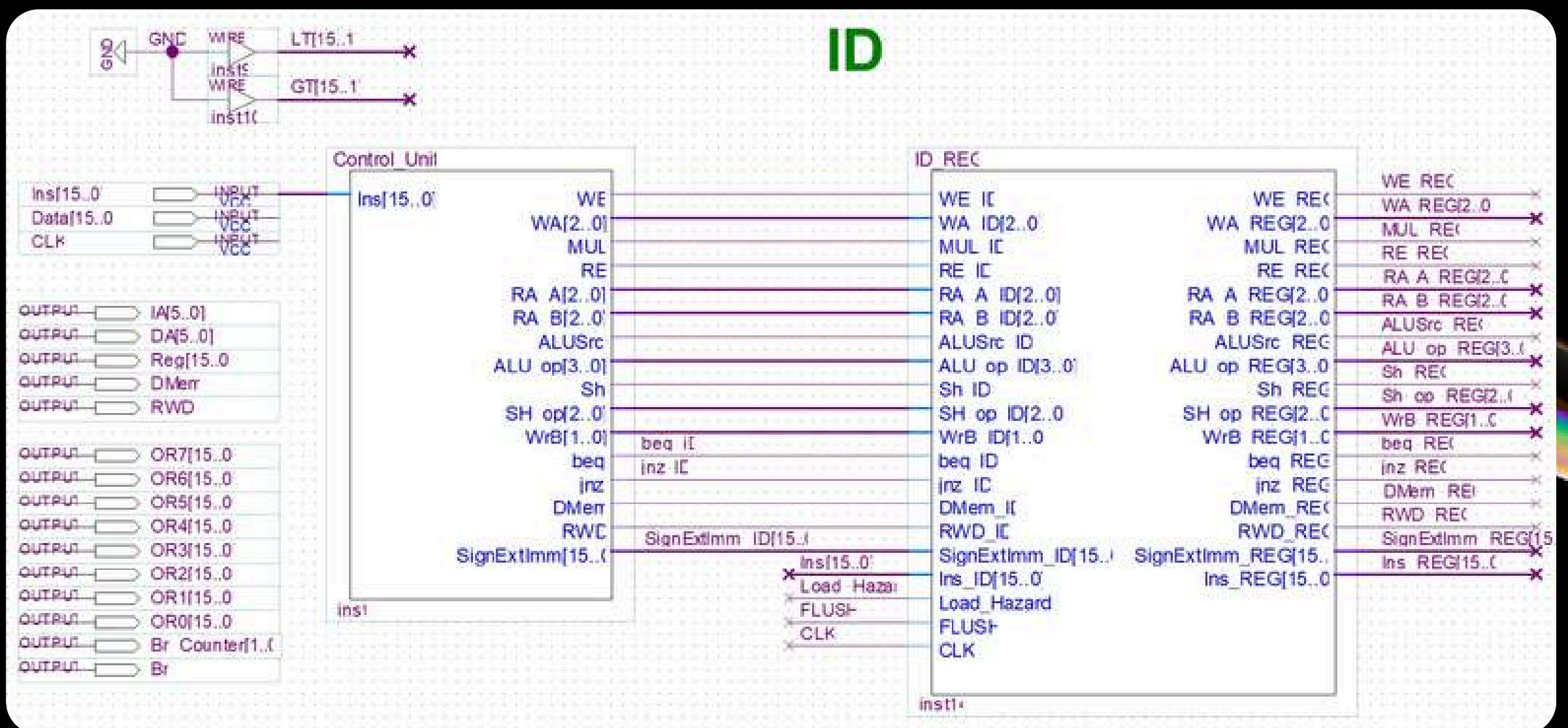
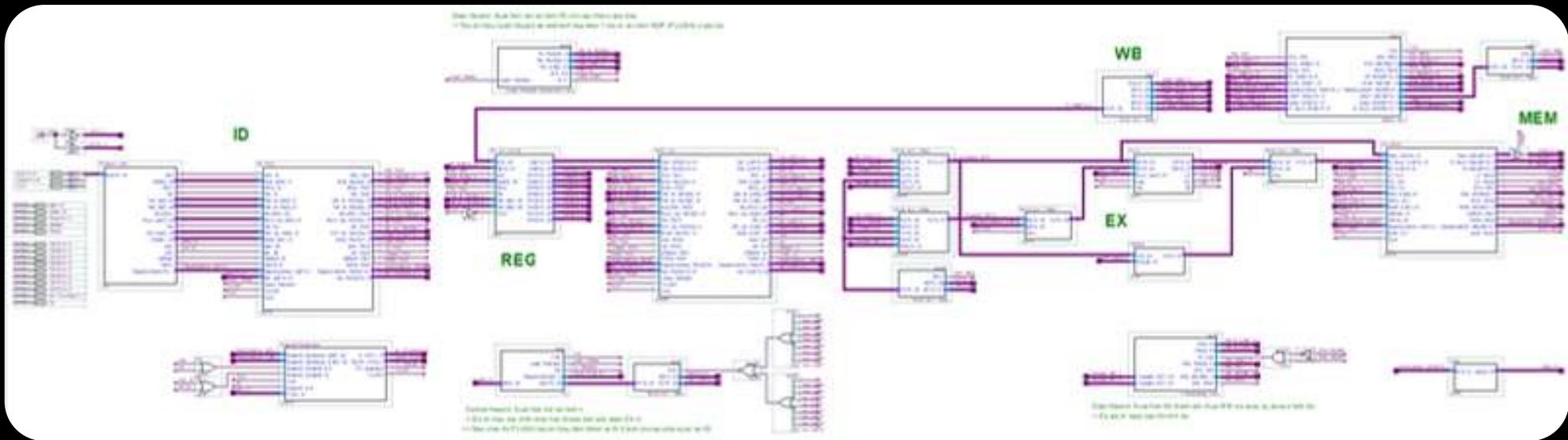


02

---

PIPELINE &  
MẠCH NHÂN

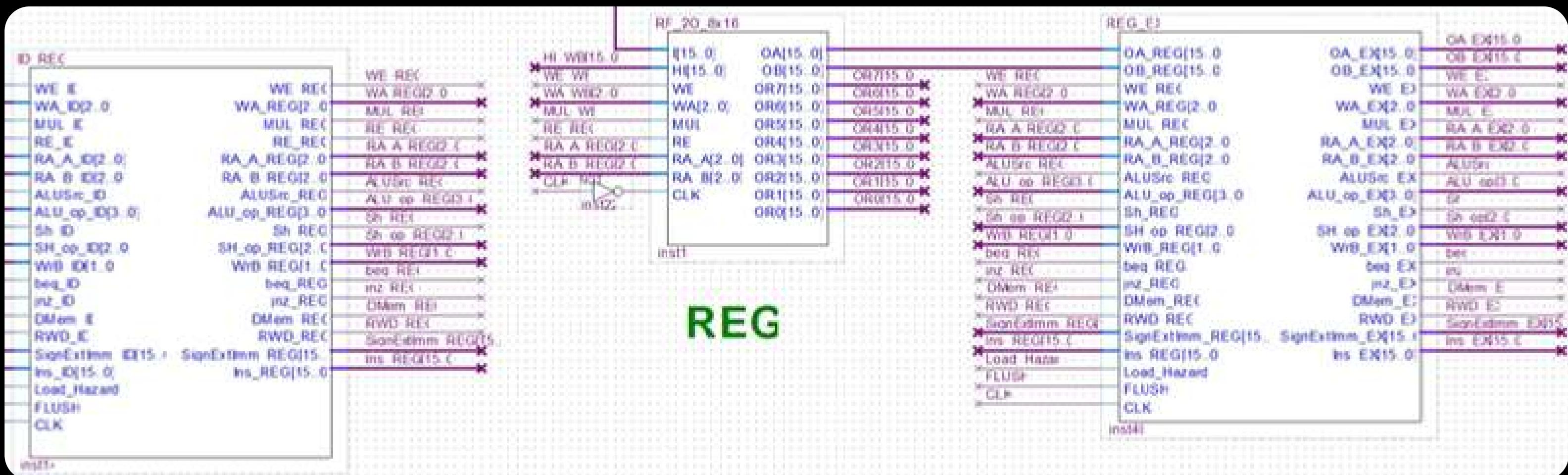
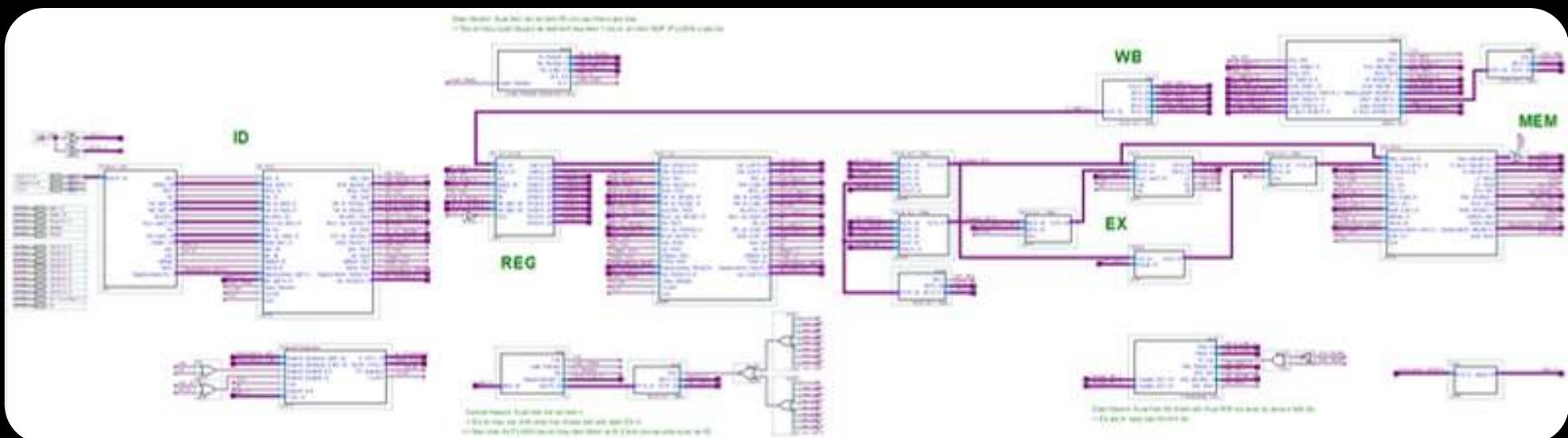
# PIPELINE - IF & ID



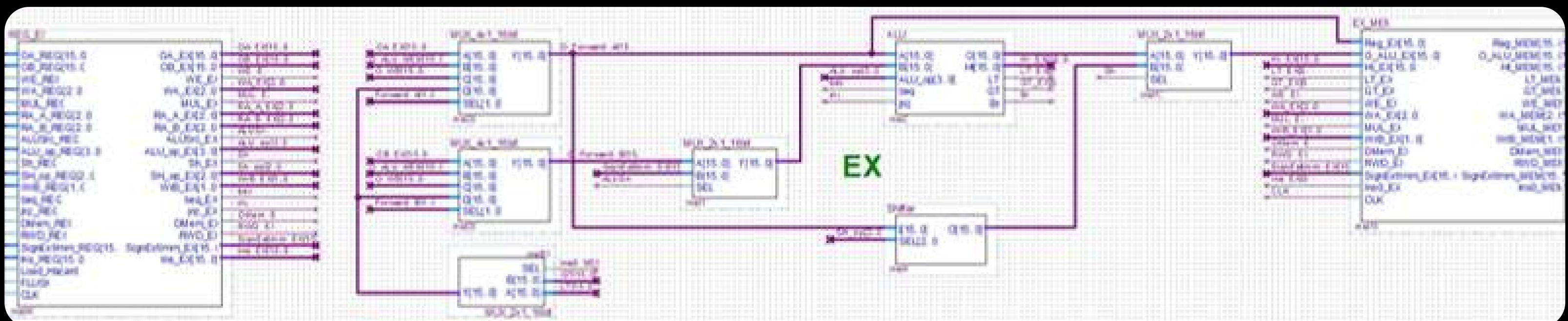
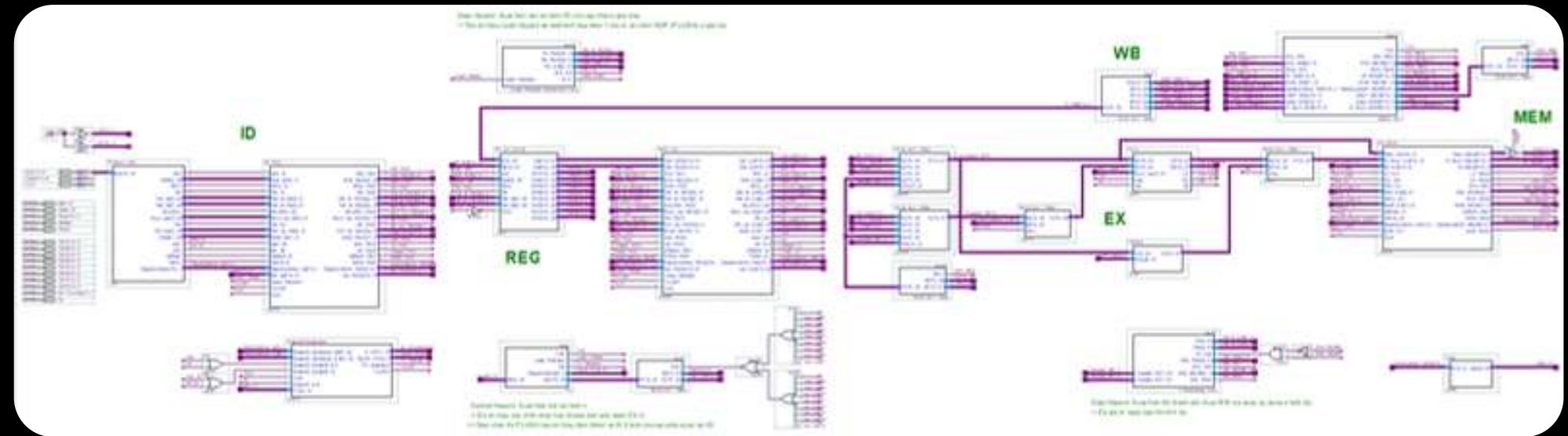
02

# PIPELINE & MÄCH NHÂN

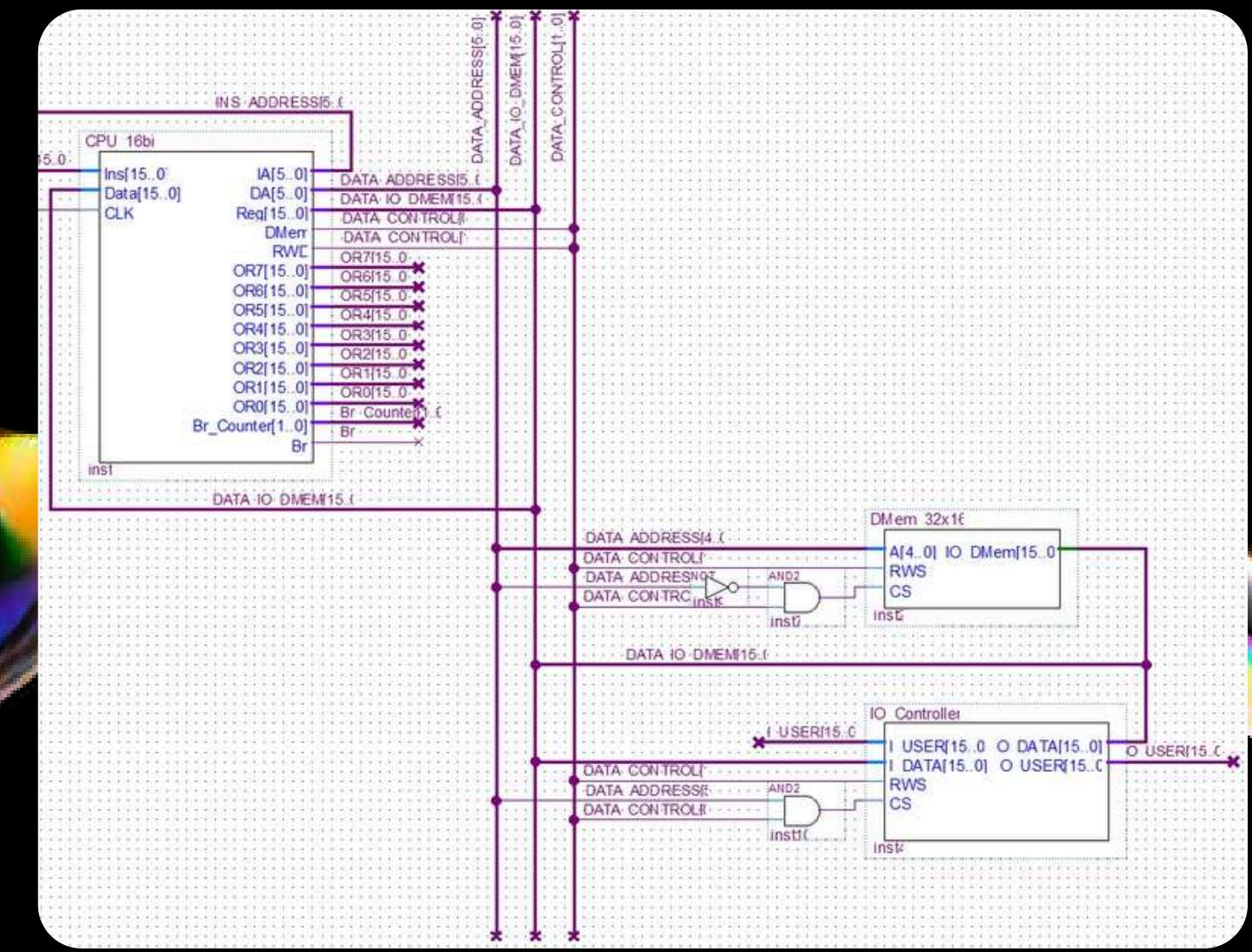
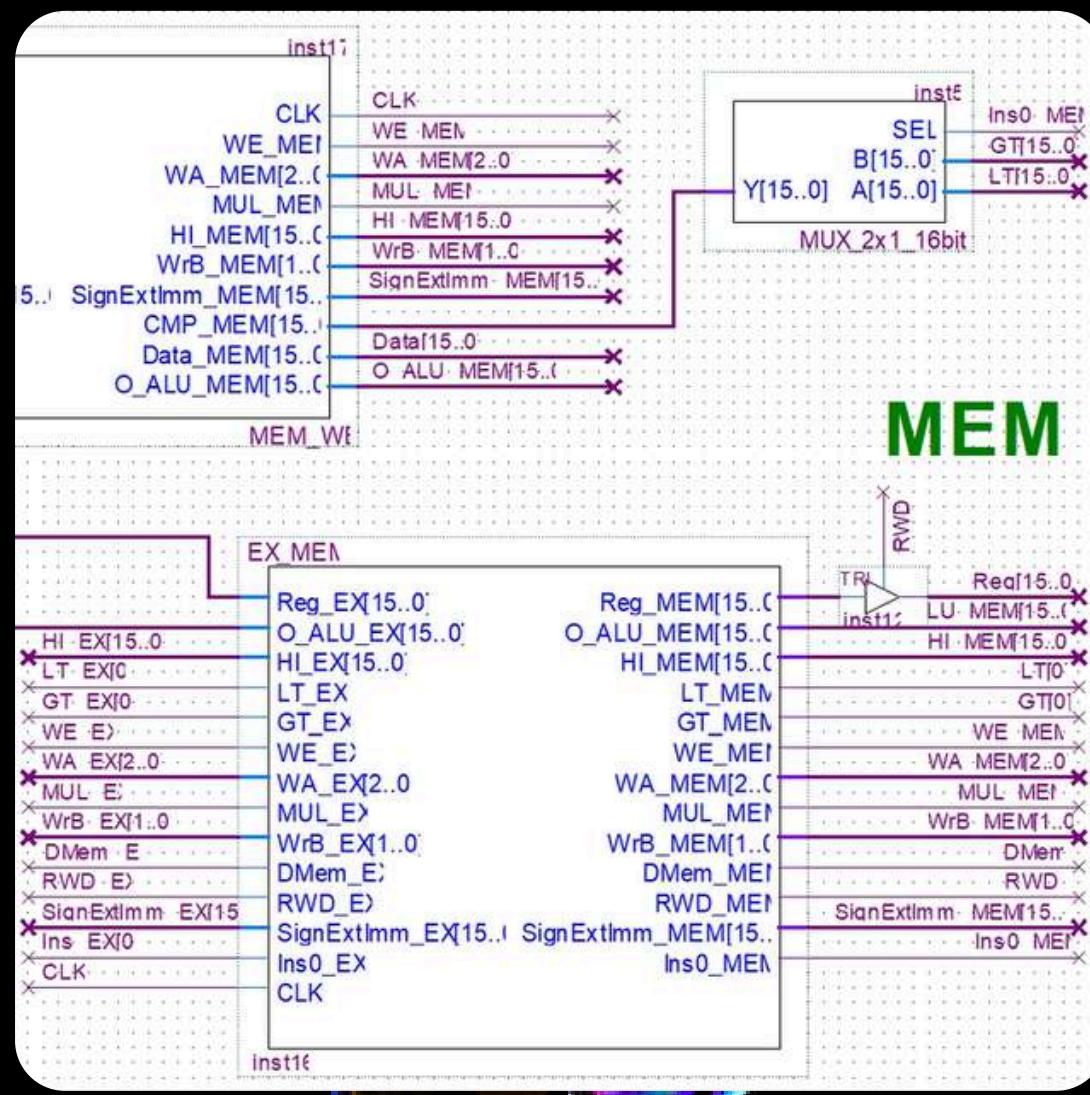
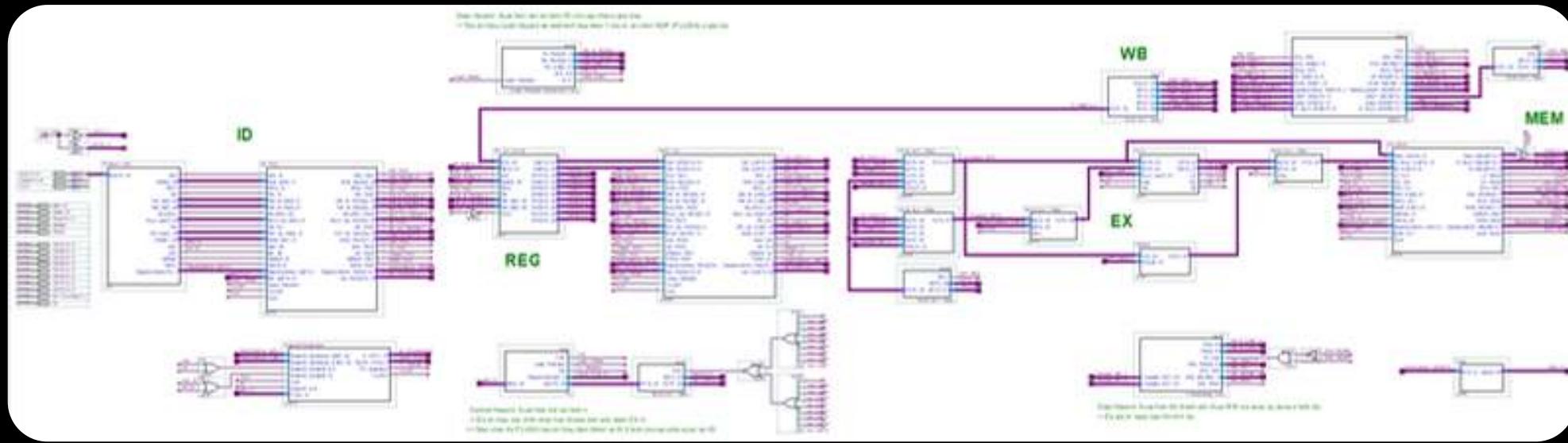
# PIPELINE - REG



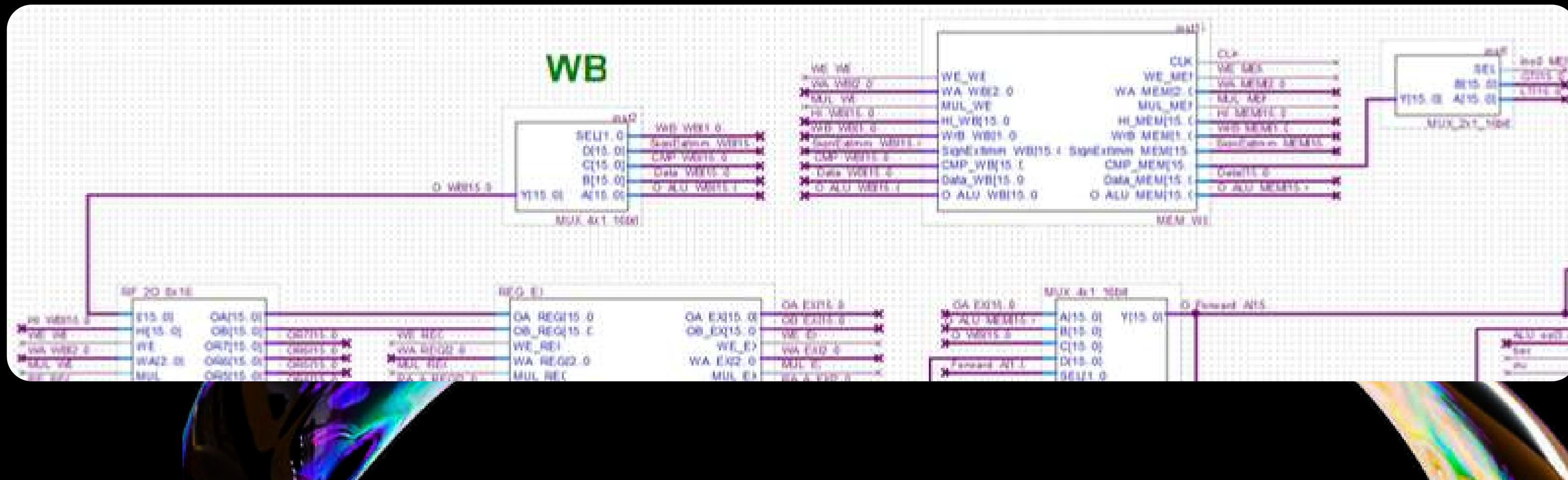
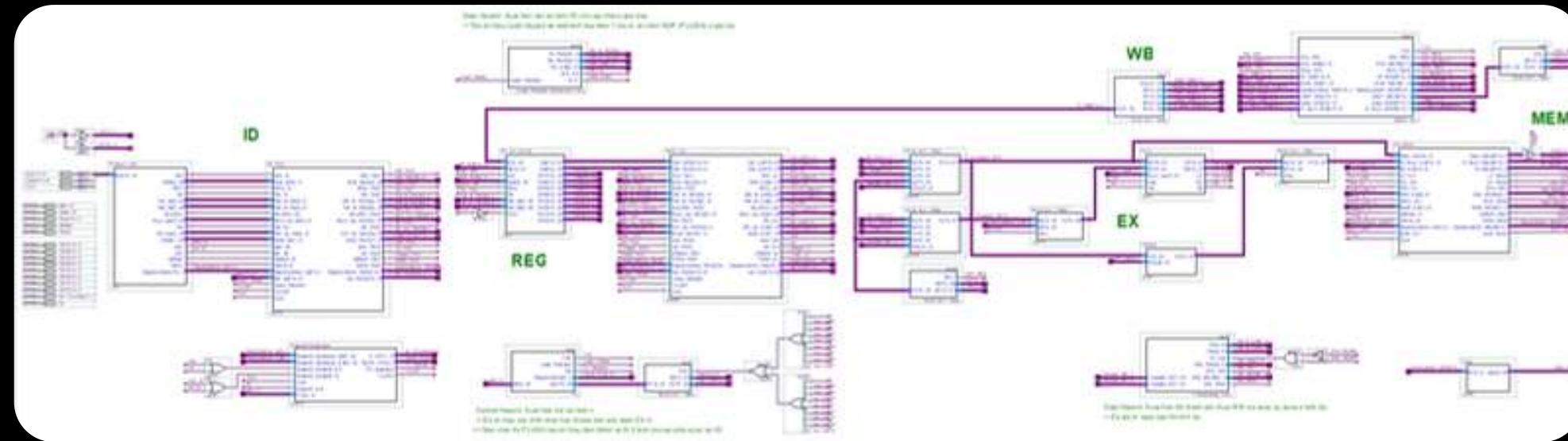
# PIPELINE - EX



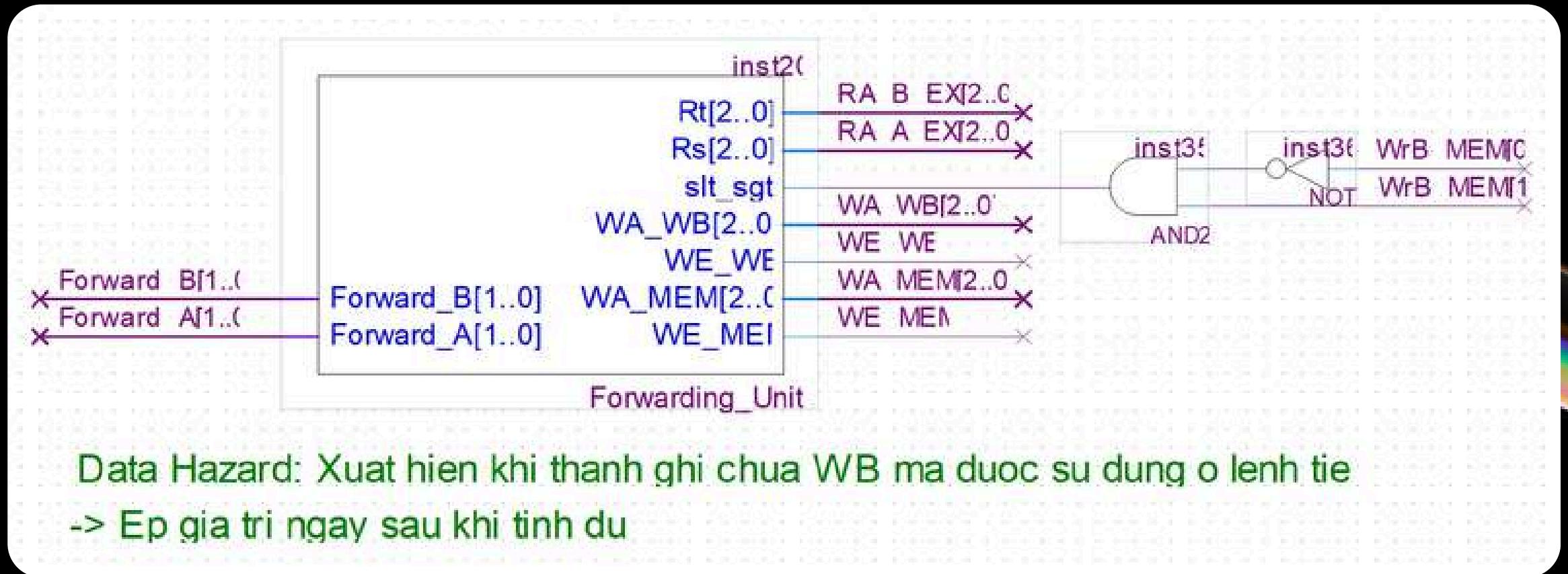
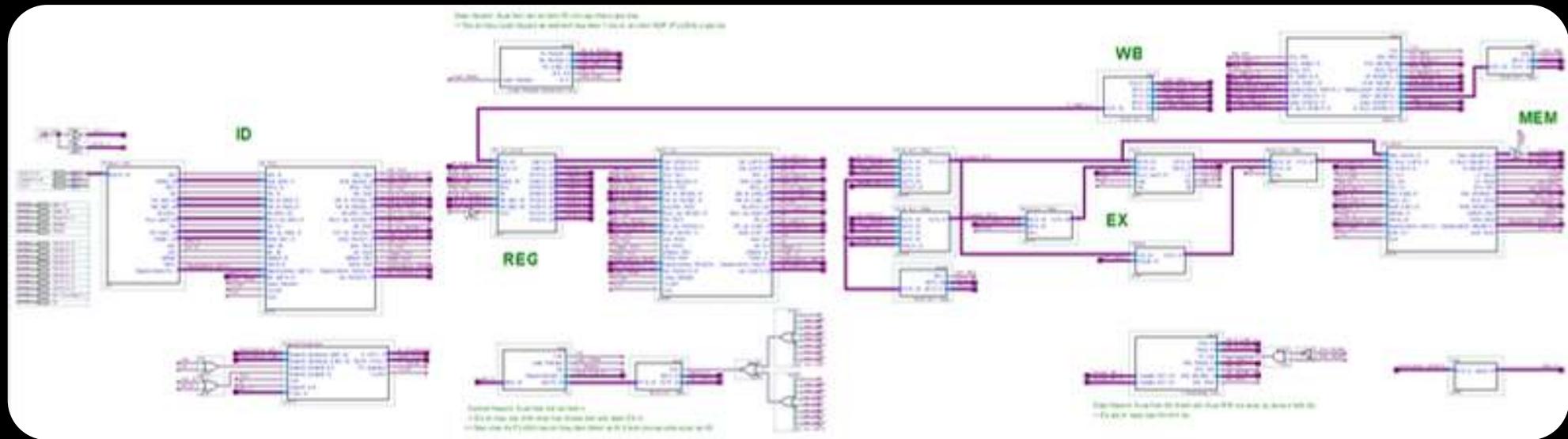
# PIPELINE - MEM



# PIPELINE - WB



# PIPELINE - FORWARDING UNIT

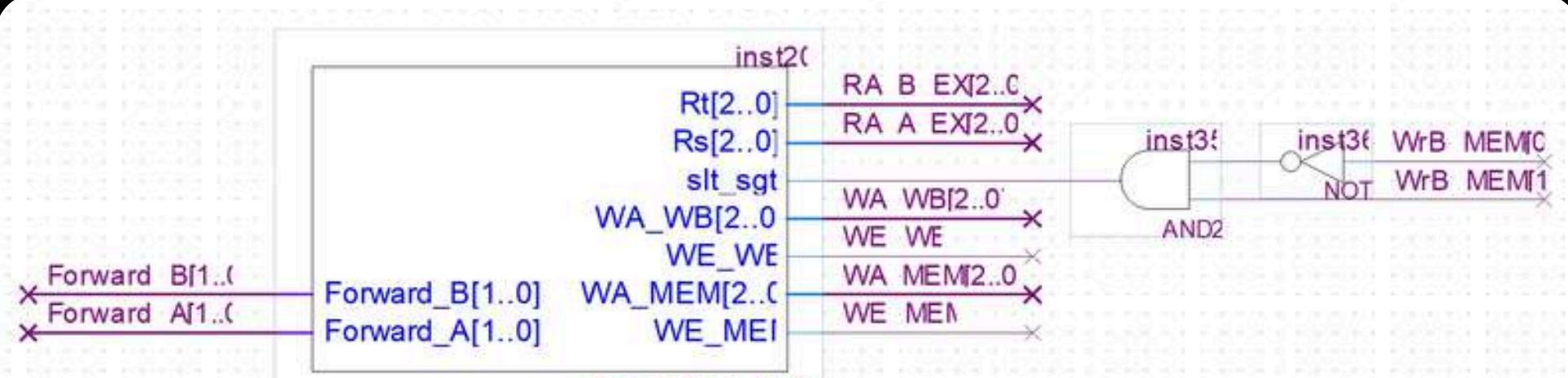


# PIPELINE - FORWARDING UNIT

```

sub    $2,    $1,    $3
and   $12,   $5,    $2      # 1b
or    $13,   $2,    $6      # 2a
add   $14,   $2,    $2      # No hazard
sw    $15, 100($2)      # No hazard

```



Data Hazard: Xuất hiện khi thanh ghi chưa WB mà được sử dụng ở lệnh tie  
-> Ep giá trị ngay sau khi tính du

```

lw    $2, 20($1)
nop
or   $8, $2, $6
add  $9, $4, $2
slt  $1, $6, $7

```

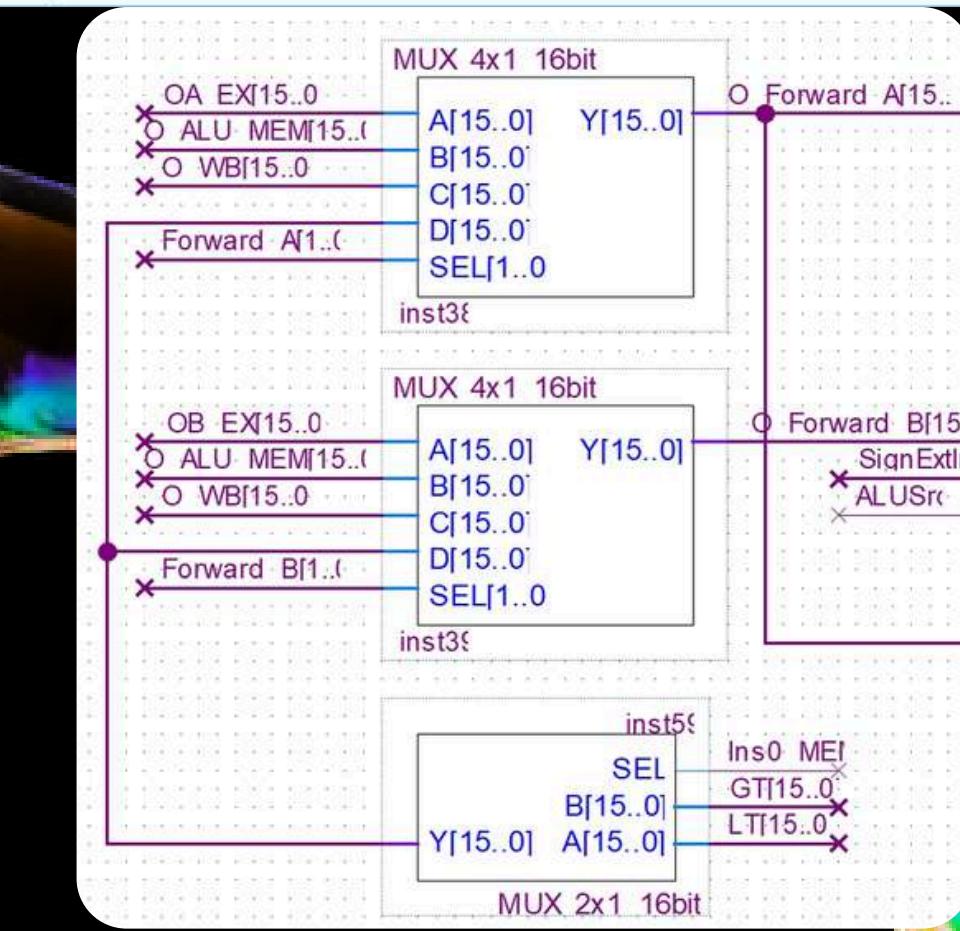
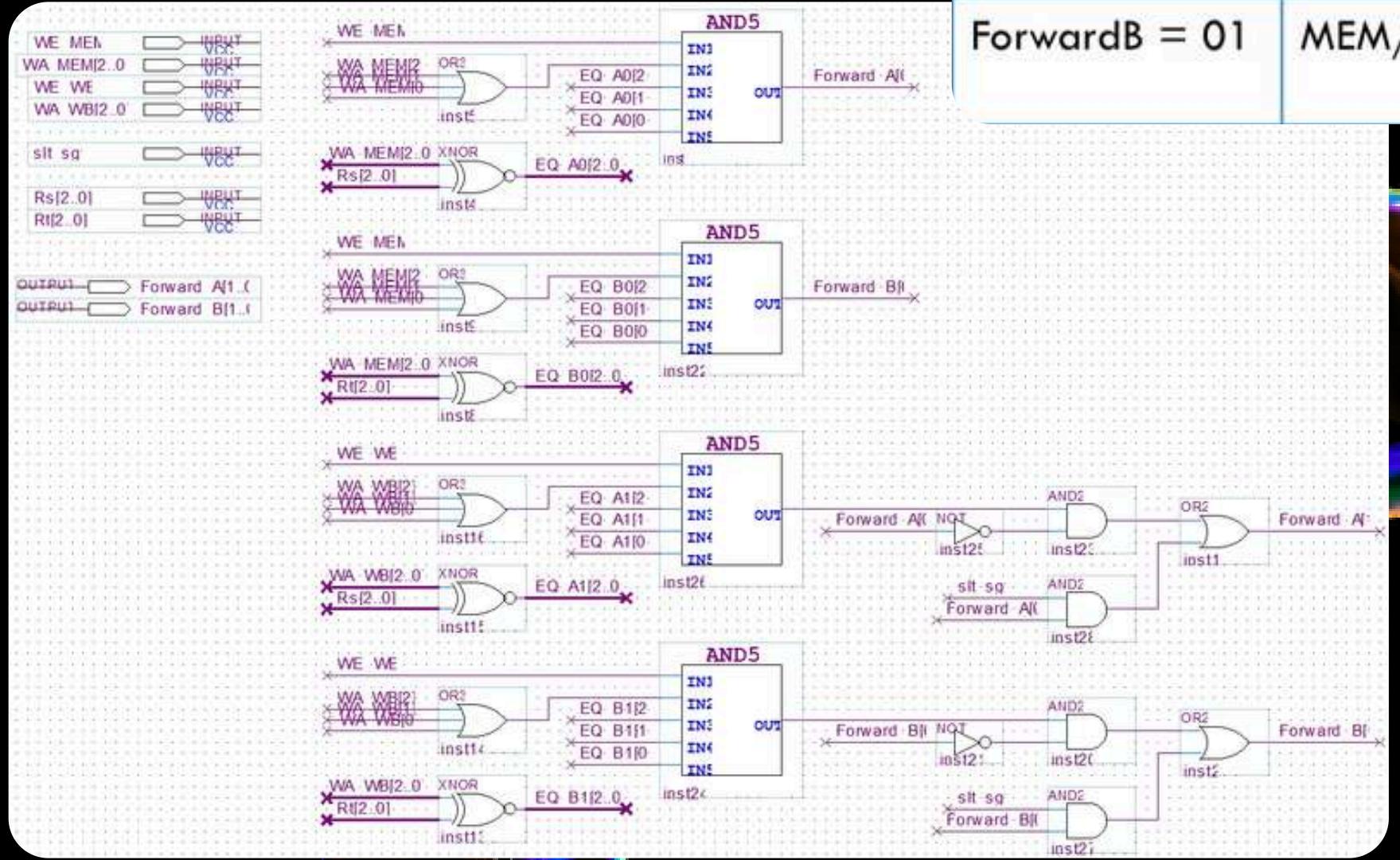
```

add  $1, $1, $2
add  $1, $1, $3
add  $1, $1, $4

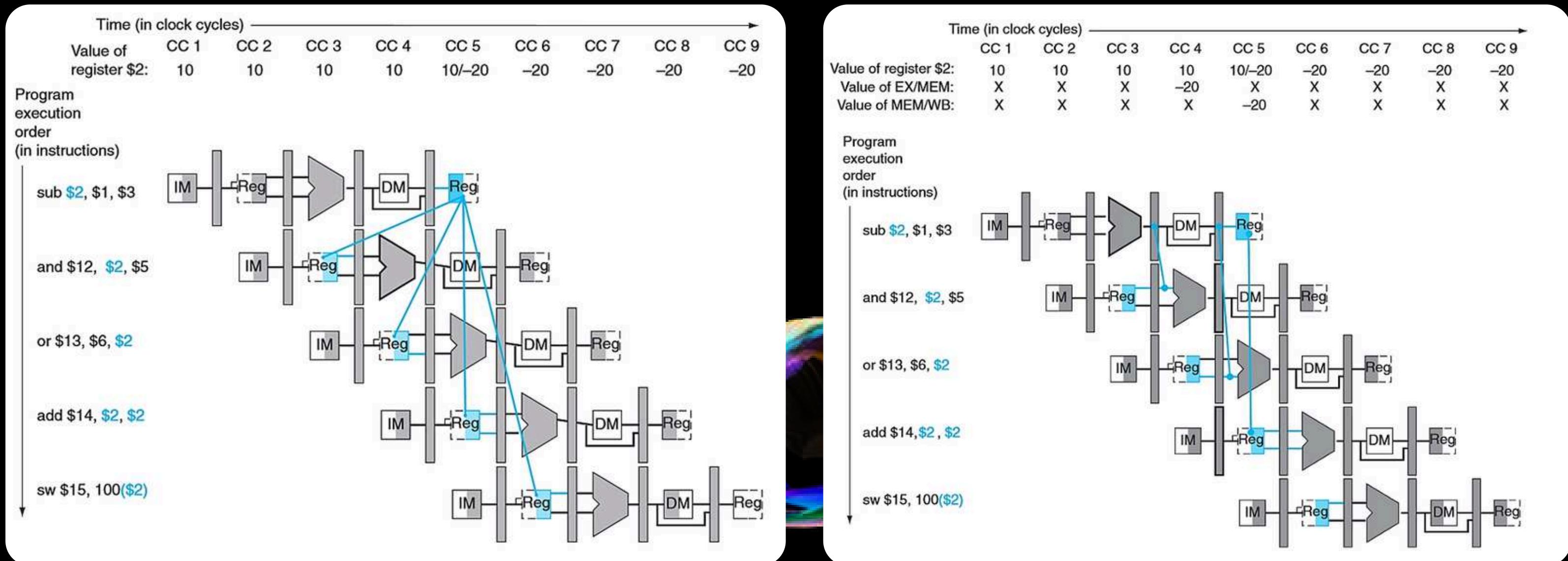
```

# PIPELINE - FORWARDING UNIT

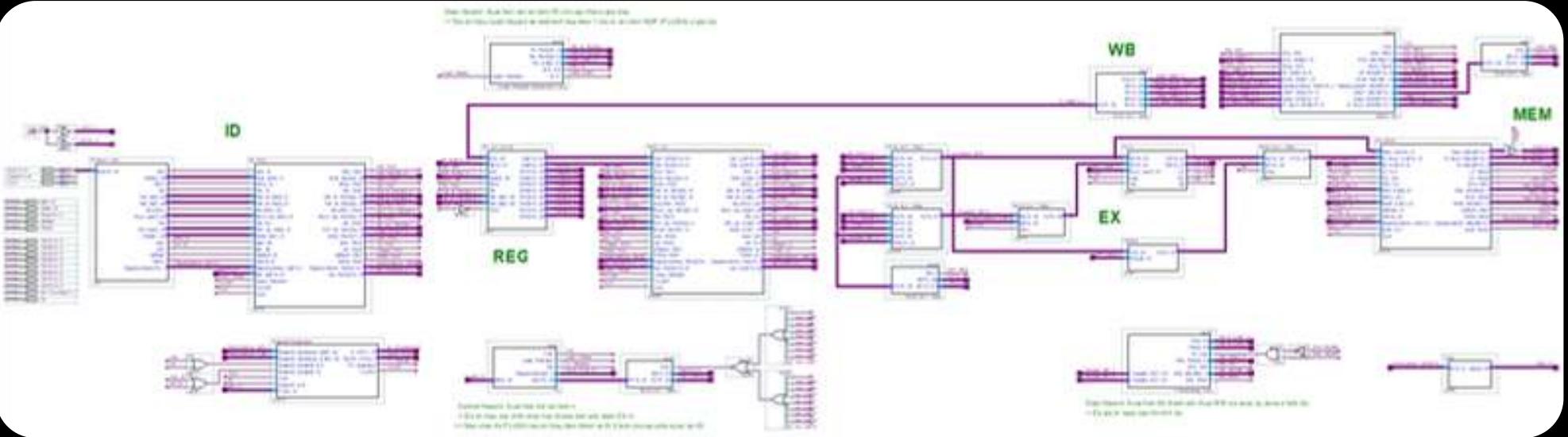
Mux Control	Source	Explanation
ForwardA = 00	ID/EX	First ALU input comes from register file.
ForwardA = 10	EX/MEM	First ALU input is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	First ALU input is forwarded from the data memory or a prior ALU result.
ForwardB = 00	ID/EX	Second ALU input comes from register file.
ForwardB = 10	EX/MEM	Second ALU input is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	Second ALU input is forwarded from the data memory or a prior ALU result.



# PIPELINE - FORWARDING UNIT

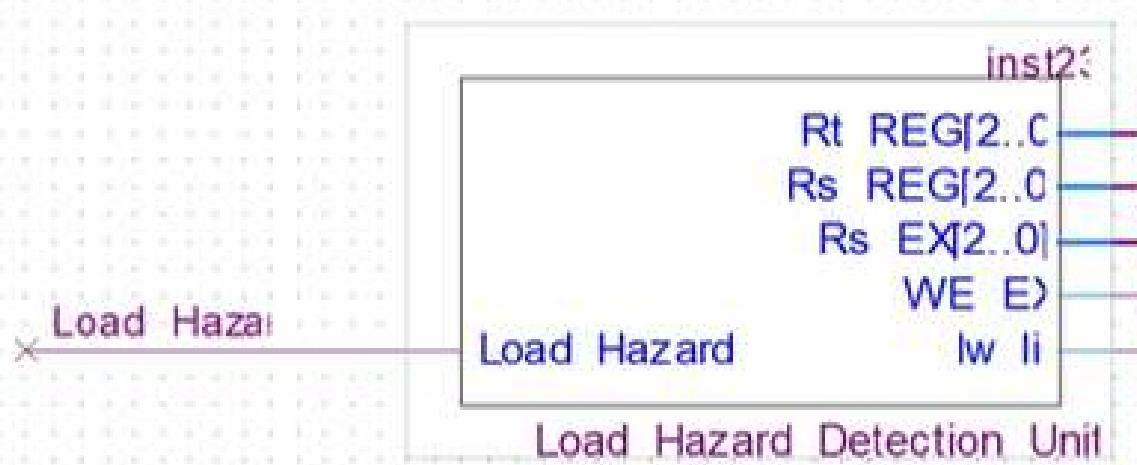


# PIPELINE - HAZARD DETECTION UNIT



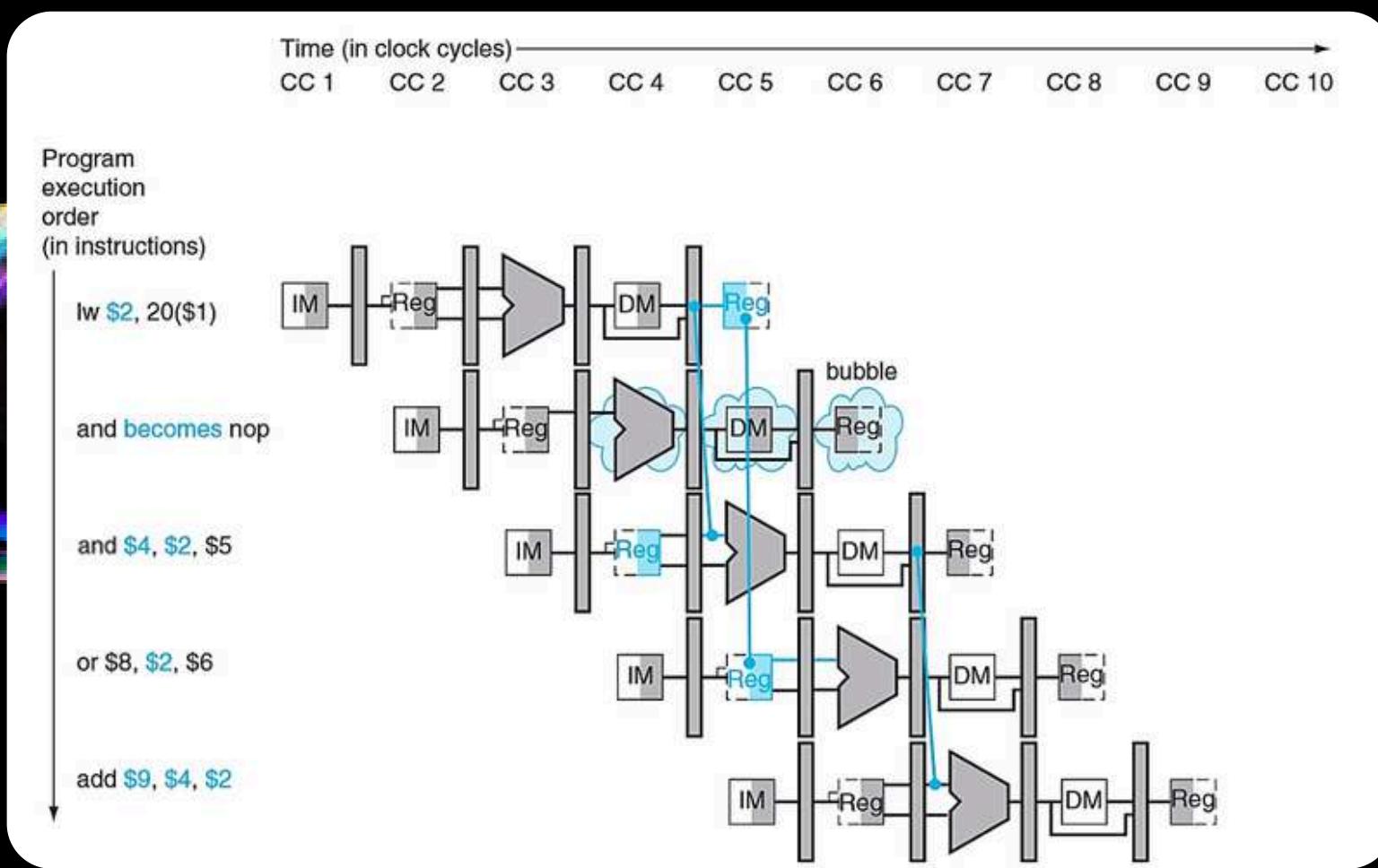
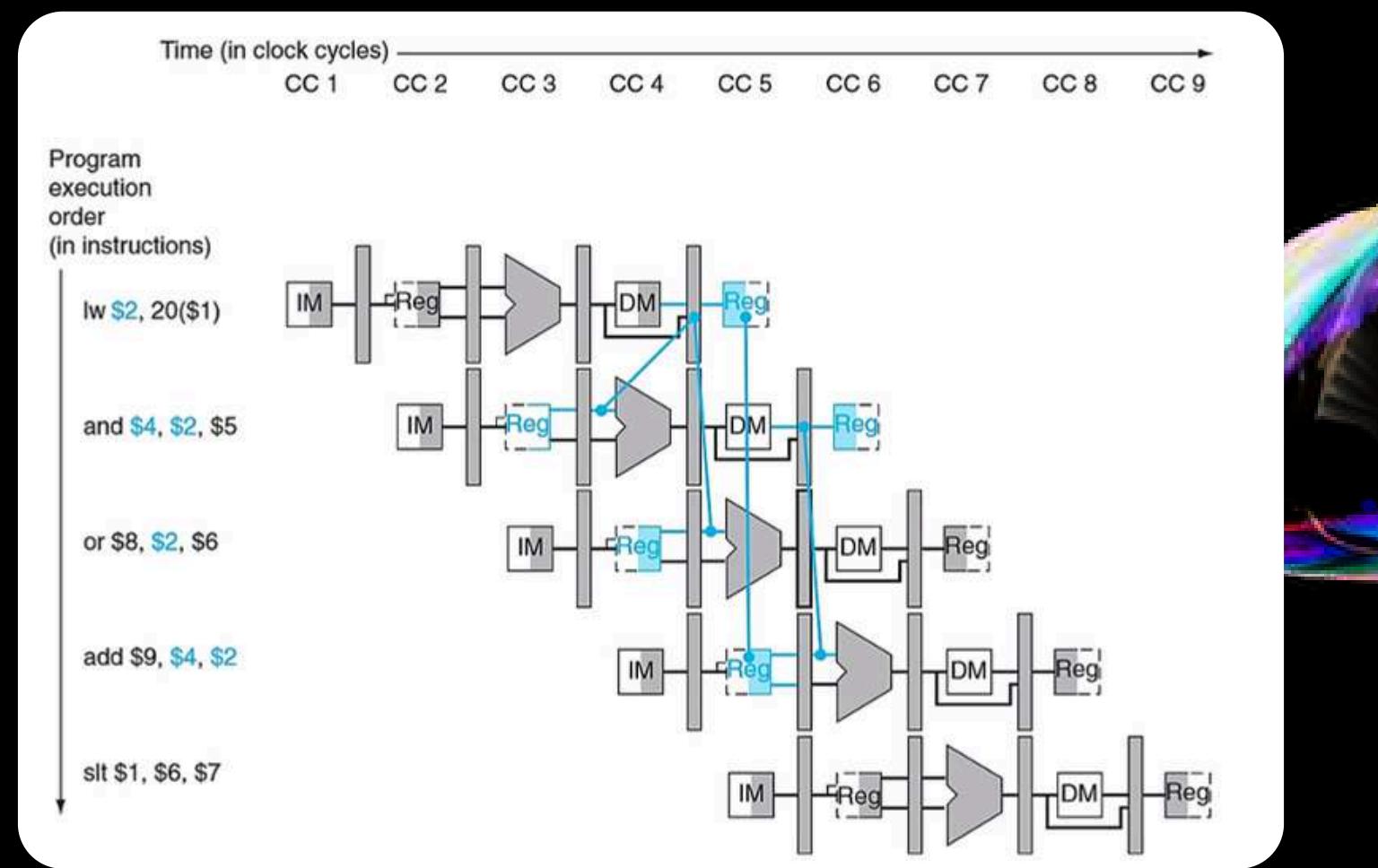
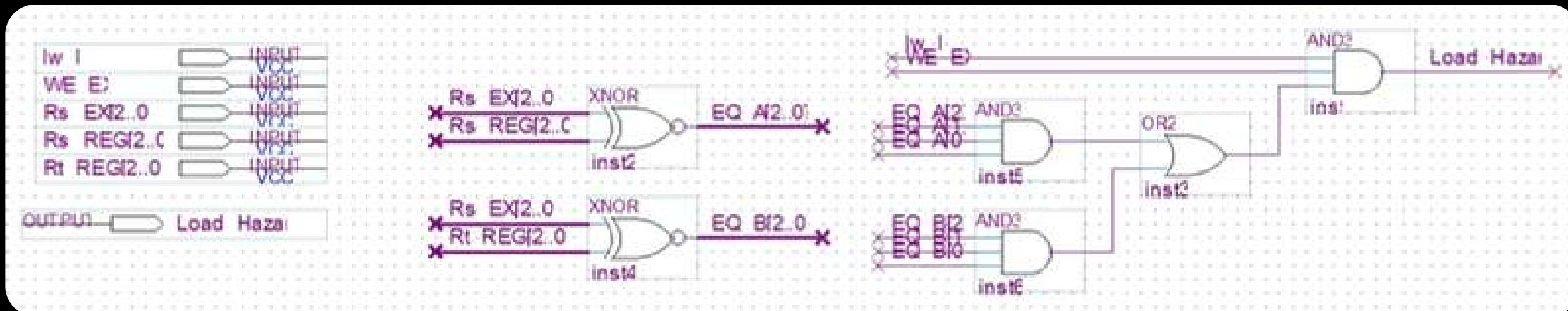
Data Hazard: Xuất hiện doi với lệnh RI chỉ cập nhật ở giao diện

-> Tao tín hiệu Load Hazard để stall lệnh tiếp theo 1 chu kỳ và chen NOP (FLUSH) hoặc

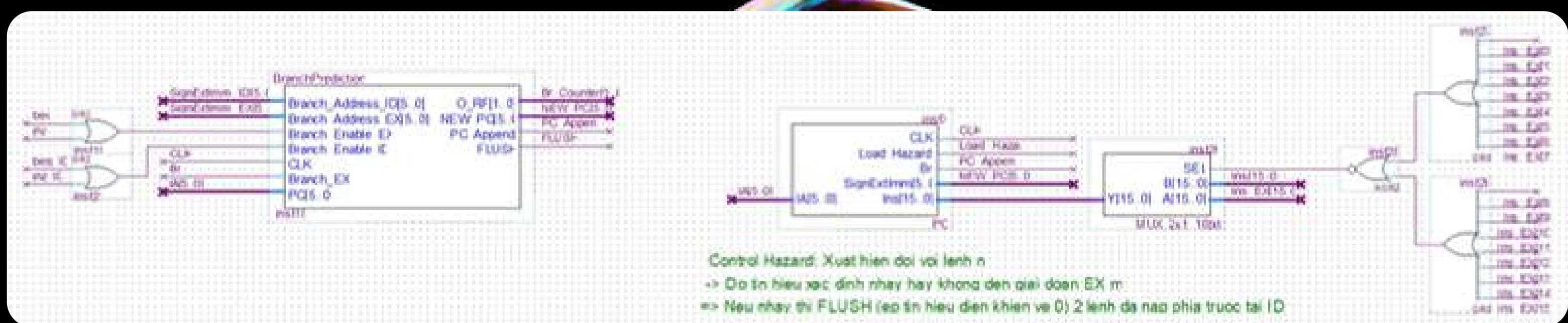
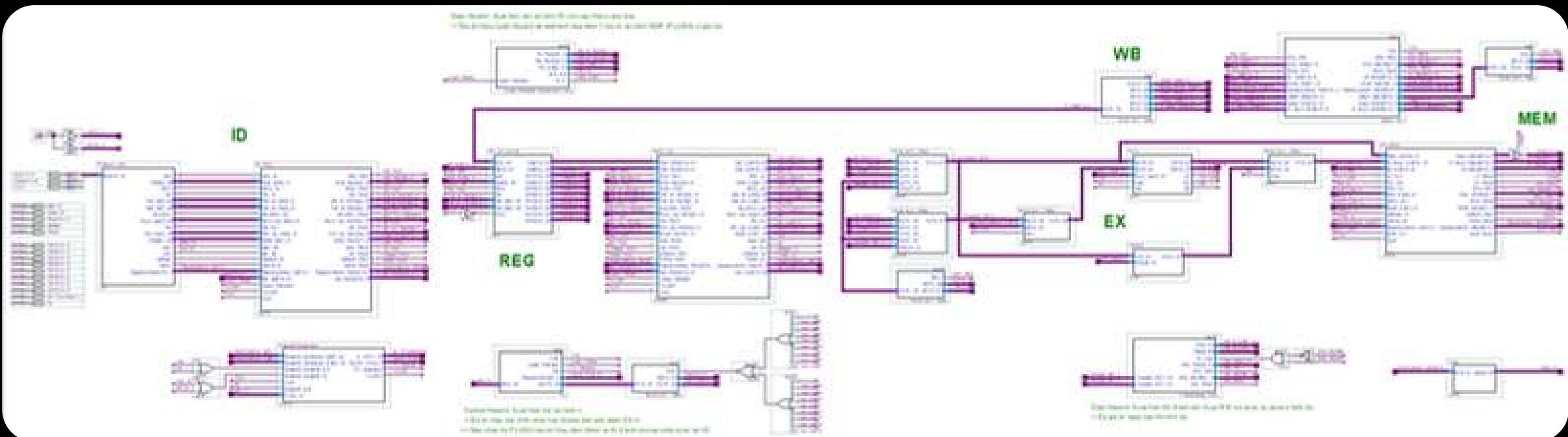


lw	\$2,	20 (\$1)
and	\$4,	\$2, \$5
or	\$8,	\$2, \$6
add	\$9,	\$4, \$2
slt	\$1,	\$6, \$7

# PIPELINE - HAZARD DETECTION UNIT



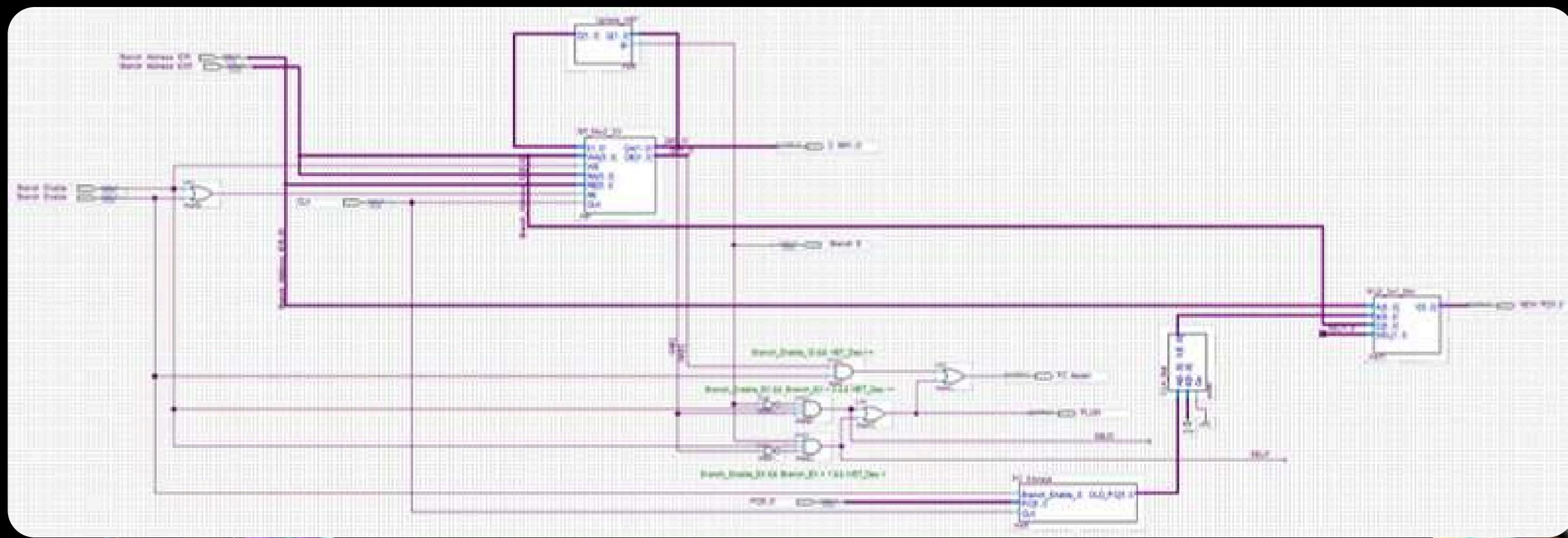
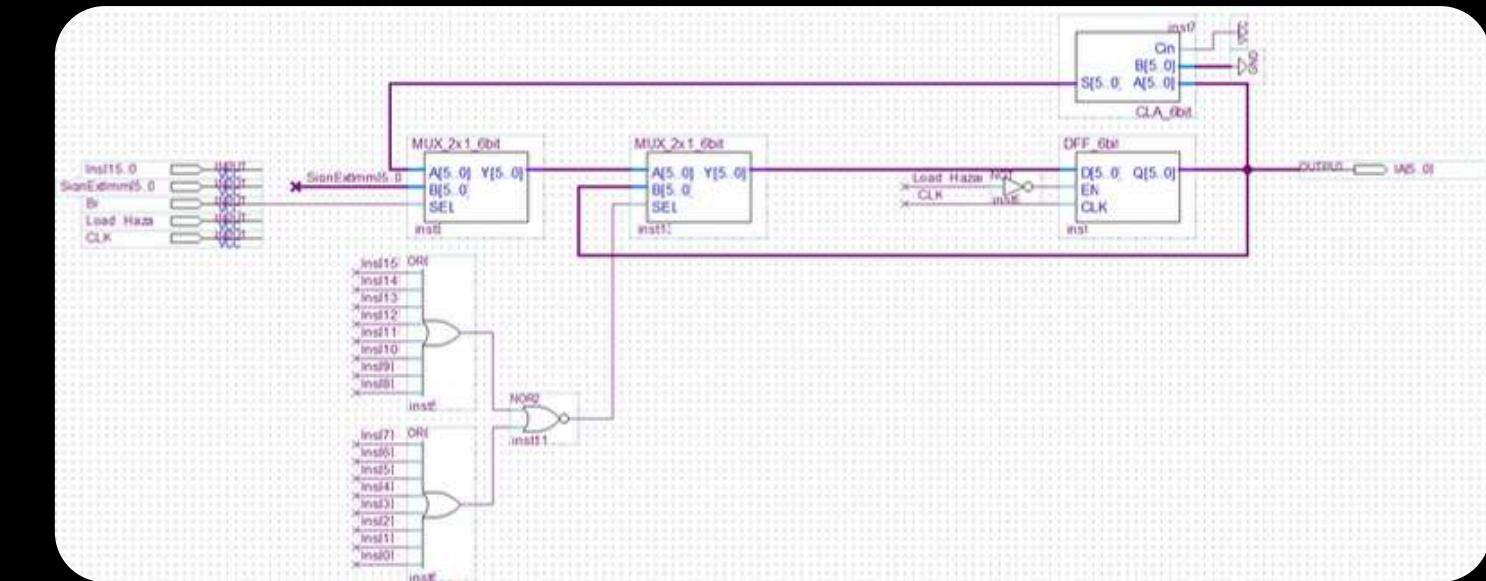
# PIPELINE - PC + BRANCH PREDICTION



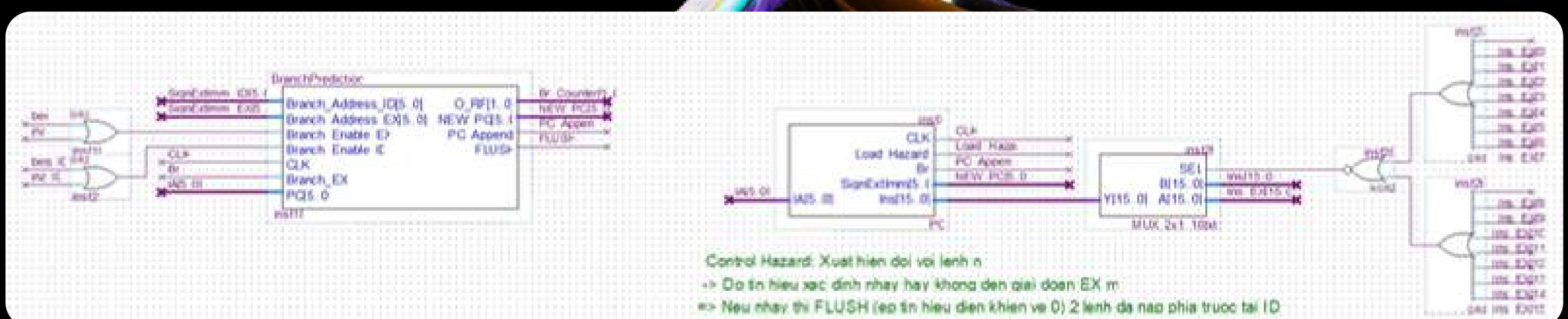
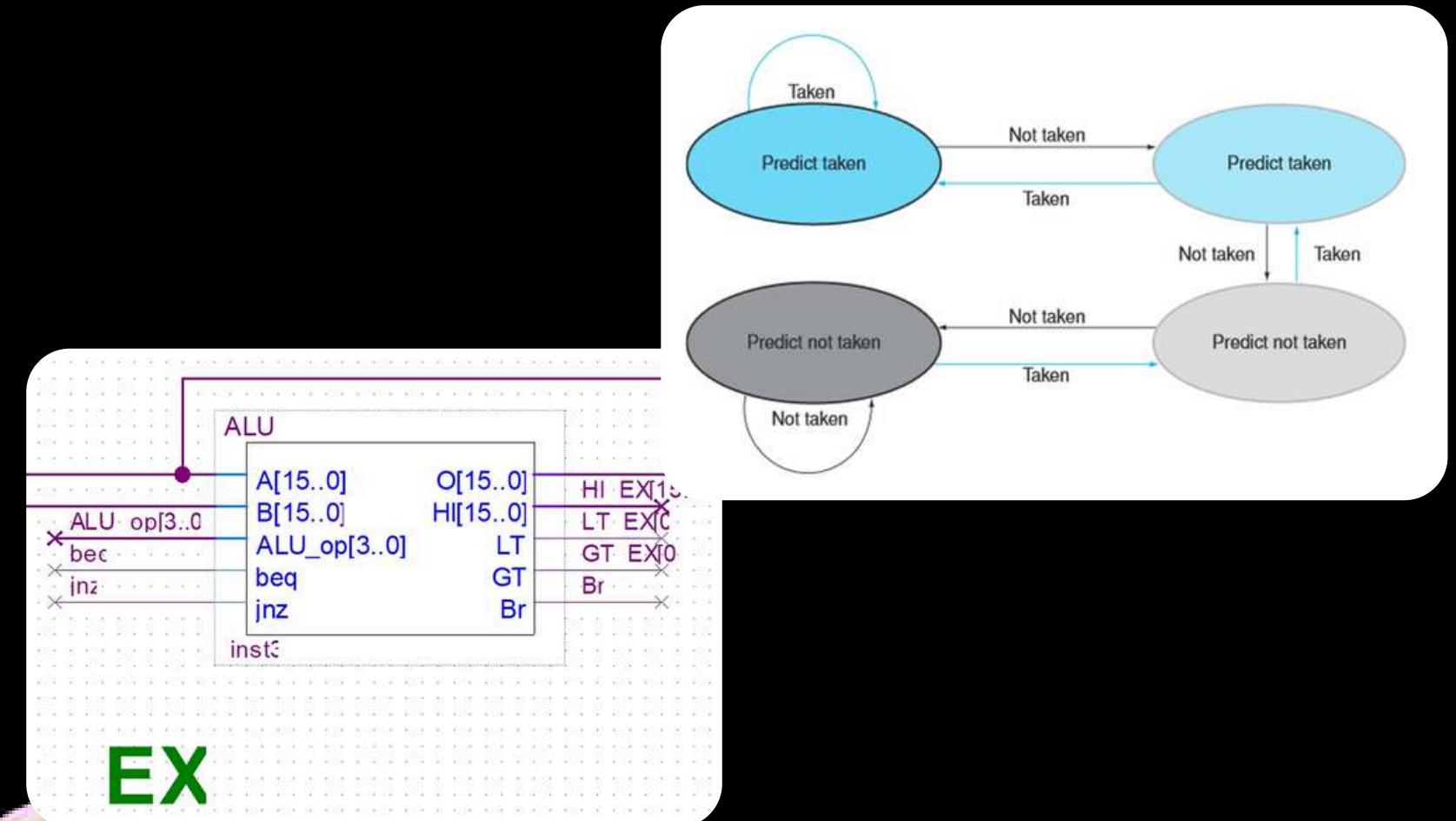
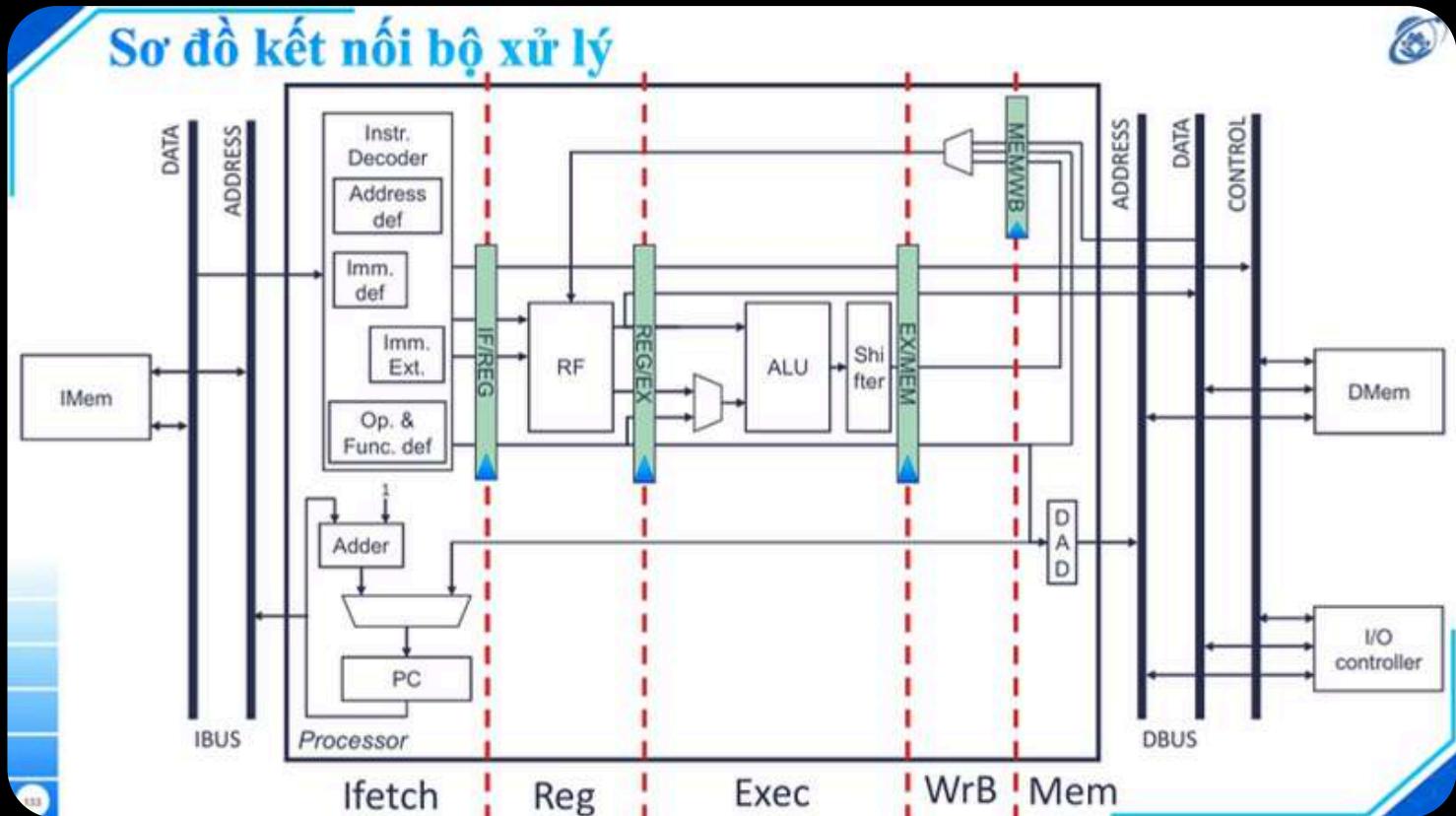
# PIPELINE - PC + BRANCH PREDICTION

```
int i = 0;
do{
    /* loop body */
    i = i + 1
}while(i < 10);
```

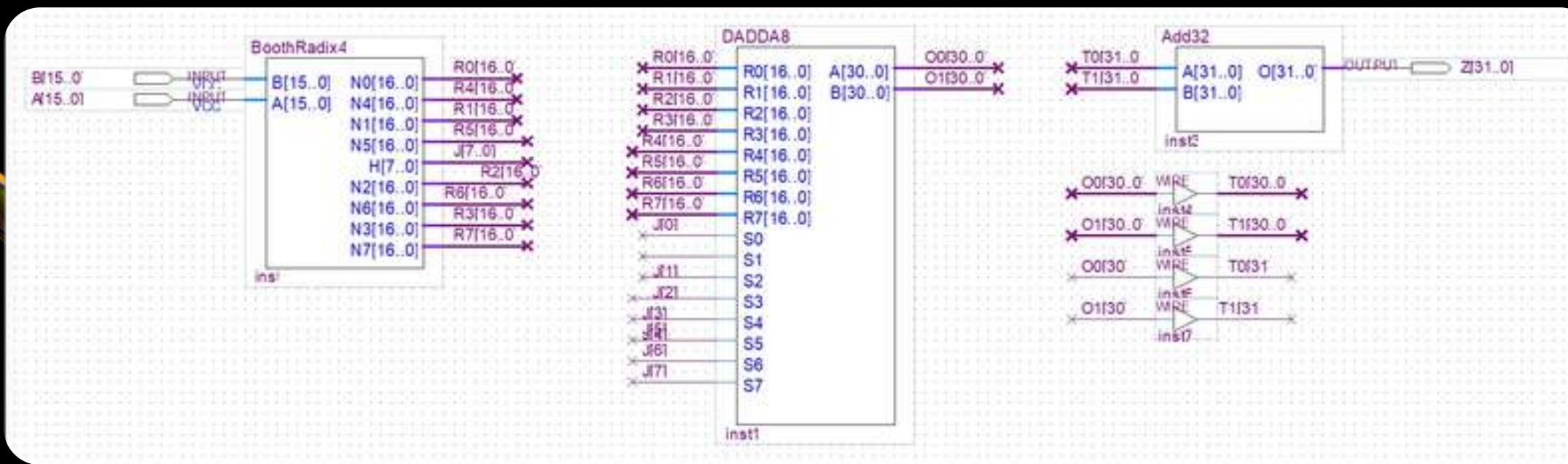
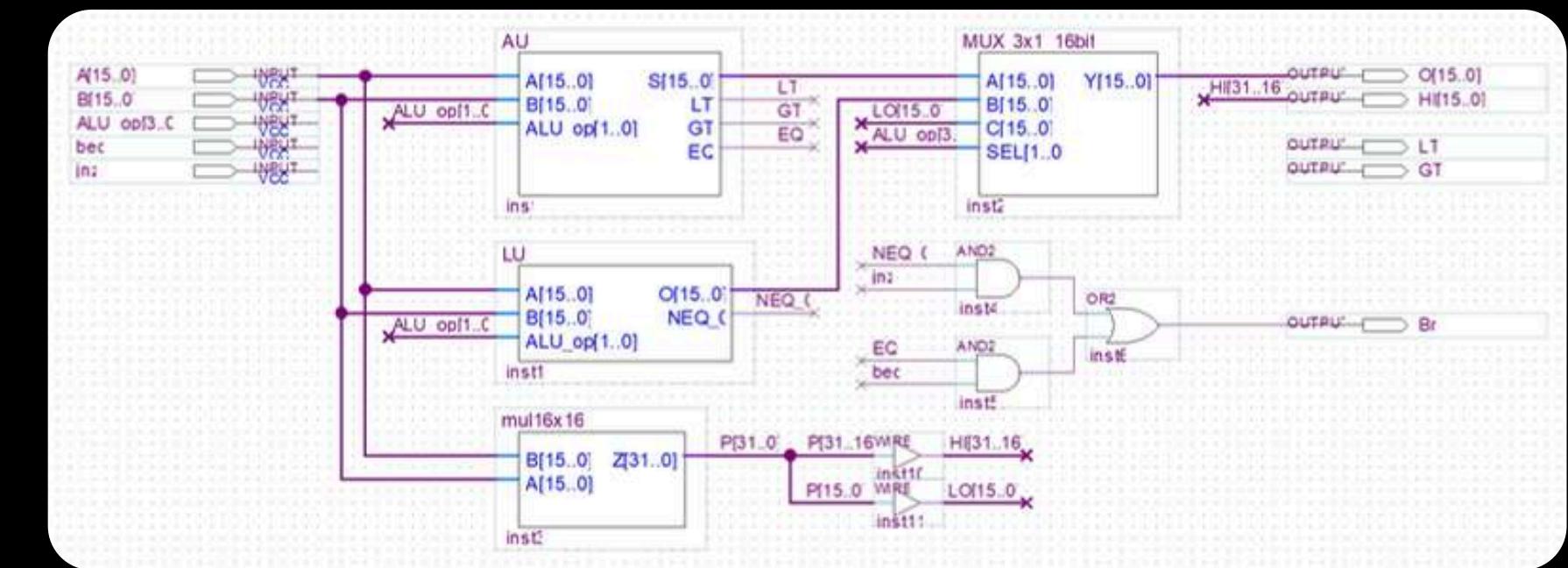
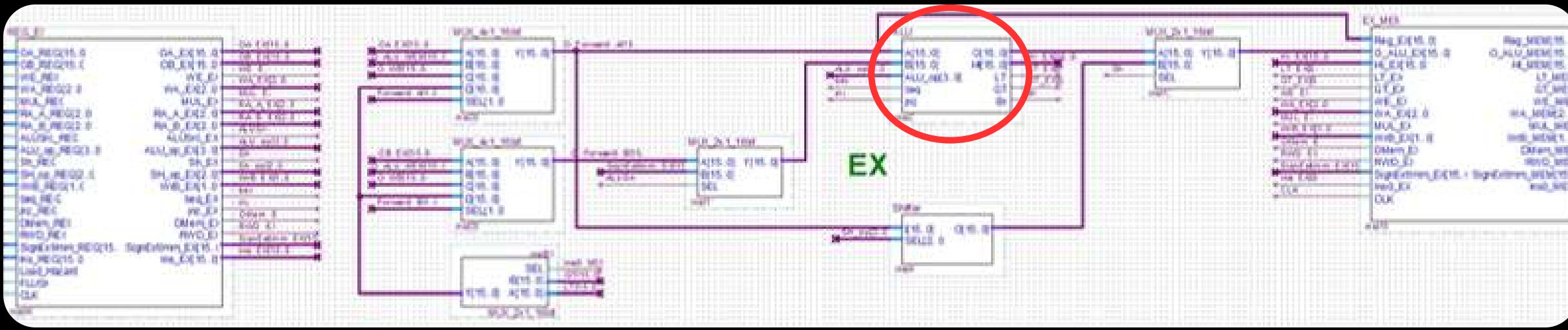
```
L1:    add    $t0, $0, $0
        /* loop body */
        addi   $t0, $t0, 1
        slti   $t1, $t0, 10
        bne    $t1, $0, L1
```



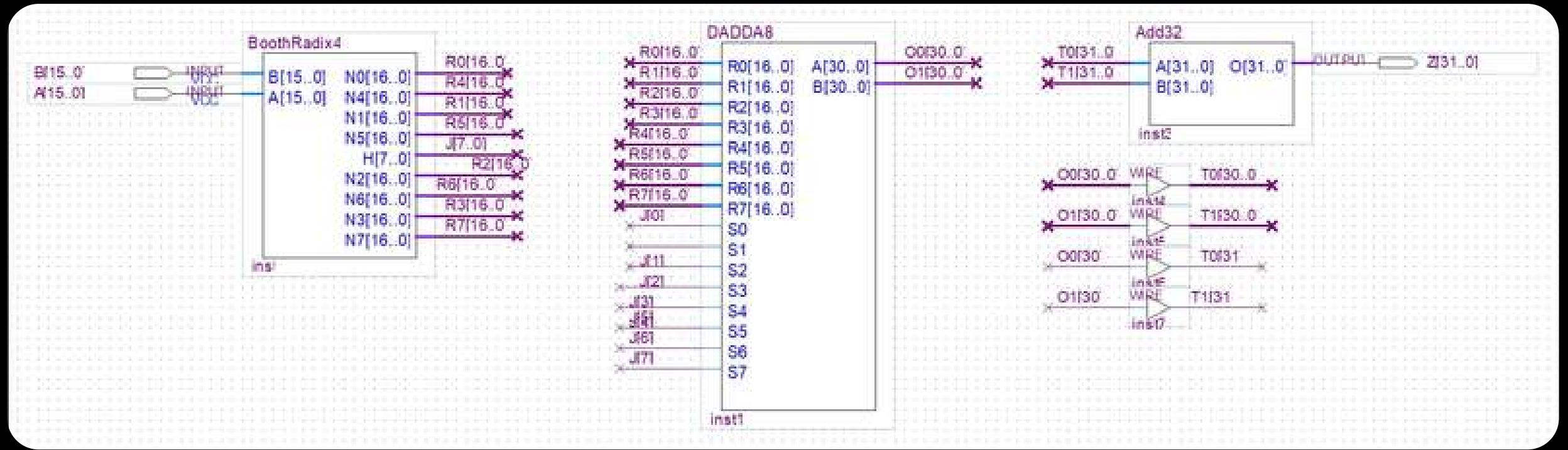
# PIPELINE - PC + BRANCH PREDICTION



# MẠCH NHÂN



# MẠCH NHÂN



→ Thuật toán Booth?

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth:  $16 - 1 = 15 \rightarrow 1$  phép toán

???

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 1111  
 $i = 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 1111 0  
 $i = 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \ -1$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 1111 0  
 $i = 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \ -1$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 11110  
 $i = 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \ -1$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 1111 0  
 $i = \begin{matrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & -1 \end{matrix}$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):       $16 - 1 = 15 \rightarrow 1$  phép toán  
                              ???

0000 **1111** 0  
 $i = \begin{matrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & -1 \end{matrix}$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):  $\textcircled{16} - \textcircled{1} = 15 \rightarrow 1$  phép toán  
???  
???

0000 **0** 1111 **1** 0  
 $i = \quad 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \quad -1$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-2):  $\textcircled{16} - \textcircled{1} = 15 \rightarrow 1$  phép toán  
????



→ Xét từ phải qua trái tại bit  $i$ :

$$+) 0 \rightarrow 1: -2^i$$

$$+) 1 \rightarrow 0: +2^i$$

⇒ Công thức Booth  
(đúng với mọi số có dấu):

$$\sum_{i=0}^{N-1} (y_{i-1} - y_i) 2^i$$

# MẠCH NHÂN

→ Cách nhân thông thường:

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \\ \times\ 0111\ 1111\ 1111\ 1111\ (32767) \\ \hline 0000\ ... 0000\ 0000\ 0000\ 1111\ (32\ bits) \end{array}$$

# MẠCH NHÂN

→ Cách nhân thông thường:

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \\ \times 0111\ 1111\ 1111\ 1111\ (32767) \\ \hline 0000 \dots 0000\ 0000\ 0000\ 1111 \\ \quad \dots \\ 0000 \dots 0000\ 0000\ 0000\ 0000 \end{array}$$

} x16

⇒ Phức tạp, không tối ưu.

# MẠCH NHÂN

→ Áp dụng Booth (Booth Radix-2):

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111 \quad (15) \rightarrow M \\ \times \quad 0111\ 1111\ 1111\ 1111\ 0 \quad (32767) \\ \hline -M << 0 \end{array}$$

# MẠCH NHÂN

→ Áp dụng Booth (Booth Radix-2):

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111 \quad (15) \rightarrow M \\ \times 0111\ 1111\ 1111\ 1111\ 0 \quad (32767) \\ \hline \end{array}$$

$$\begin{array}{r} -M \ll 0 \\ 0000 \dots 0000\ 0000\ 0000 \\ \dots \\ 0000 \dots 0000\ 0000\ 0000 \\ M \ll 15 \end{array}$$

x16

⇒ Giảm được độ phức tạp khi cộng nhưng chưa hoàn toàn tối ưu.  
Độ phức tạp không giảm nếu chuỗi bit thay đổi liên tục (TH xấu nhất).

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-4):  $16 - 1 = 15 \rightarrow 1$  phép toán

0000 1111 0  
i = 7 6 5 4 3 2 1 0 -1

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-4):  $16 - 1 = 15 \rightarrow 1$  phép toán

0000 **1111** 0  
 $i = \begin{matrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & -1 \end{matrix}$

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-4):  $\textcircled{16} - \textcircled{1} = 15 \rightarrow 1$  phép toán



# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

Booth (Radix-4):  $16 - 1 = 15 \rightarrow 1$  phép toán



Booth (Radix-2):

$$\sum_{i=0}^{N-1} (y_{i-1} - y_i) 2^i$$

$\Rightarrow$

Booth (Radix-4):

$$\sum_{i=0}^{N/2-1} (-2y_{2i+1} + y_{2i} + y_{2i-1}) \cdot 4^i$$

# MẠCH NHÂN

→ Áp dụng Booth (Booth Radix-4):

$$\begin{array}{r} 0000 \ 0000 \ 0000 \ 1111 \quad (15) \rightarrow M \\ \times \ 0111 \ 1111 \ 1111 \ 1111 \ 0 \quad (32767) \\ \hline -M << 0 \end{array}$$

# MẠCH NHÂN

→ Áp dụng Booth (Booth Radix-4):

$$\begin{array}{r}
 \times 0000\ 0000\ 0000\ 1111 \quad (15) \rightarrow M \\
 \times 0111\ 1111\ 1111\ 1111\ 0 \quad (32767) \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 -M \ll 0 \\
 0000 \dots 0000\ 0000\ 0000\ 0000 \\
 \dots \\
 0000 \dots 0000\ 0000\ 0000\ 0000 \\
 +2M \ll 14
 \end{array}$$

x8 (Sử dụng Dadda để cộng đồng thời 8 hàng)

⇒ Mức độ tối ưu cao.  
Thuật toán dễ dàng thiết kế.



03

KIỂM ĐỊNH

# KIỂM TRA TÀI NGUYÊN

## Analysis & Synthesis Resource Usage Summary

	Resource	Usage
1	Estimated Total logic elements	3,128
2		
3	Total combinational functions	2793
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	1851
2	-- 3 input functions	842
3	-- <=2 input functions	100
4		
5	Logic elements by mode	
1	-- normal mode	2793
2	-- arithmetic mode	0
3		
4	Total registers	541
5	-- Dedicated logic registers	541
6	-- I/O registers	0
7		
8	I/O pins	210
9	Embedded Multiplier 9-bit elements	0
10	Maximum fan-out node	CLK
11	Maximum fan-out	541
12	Total fan-out	11779
13	Average fan-out	3.32

Object Navigator	Entity	Logic Cells	Dedicated Logic Registers	I/O Registers	Memory Bits	HAKs	DSP Elements	DSP Shift	DSP 1bit/S	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	out/Register LCs
Cyclone II EPICMP470C														
-> Datasheet		2999 (0)	451 (0)	0 (0)	0	0	0	0	0	210	0	2999 (0)	79 (0)	345 (0)
-> CPU_Initial		1956 (41)	451 (0)	0 (0)	0	0	0	0	0	0	0	1956 (41)	79 (0)	345 (0)
-> CPU_Clock		9 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	9 (0)	0 (0)	1 (0)
-> CPU_0x1_16bit_m00		189 (0)	189 (0)	0 (0)	0	0	0	0	0	0	0	189 (0)	0 (0)	189 (0)
-> CPU_0x1_16bit_m01		10 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	10 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m02		839 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	839 (0)	0 (0)	839 (0)
-> CPU_0x1_16bit_m03		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m04		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m05		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m06		189 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	189 (0)	0 (0)	189 (0)
-> CPU_0x1_16bit_m07		9 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	9 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m08		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m09		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m10		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m11		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m12		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m13		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m14		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m15		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m16		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m17		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m18		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m19		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m20		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m21		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m22		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m23		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m24		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m25		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m26		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m27		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m28		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m29		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m30		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m31		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m32		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m33		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m34		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m35		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m36		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m37		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m38		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m39		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m40		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m41		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m42		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m43		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m44		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m45		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m46		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m47		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m48		14 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (0)	0 (0)	0 (0)
-> CPU_0x1_16bit_m49		14 (0)	0 (0)	0 (0)	0	0								

# KIỂM TRA ĐỊNH THỜI

## Slow Model Fmax Summary

	Emax	Restricted Emax	Clock Name	Note
1	60.14 MHz	60.14 MHz	clk	

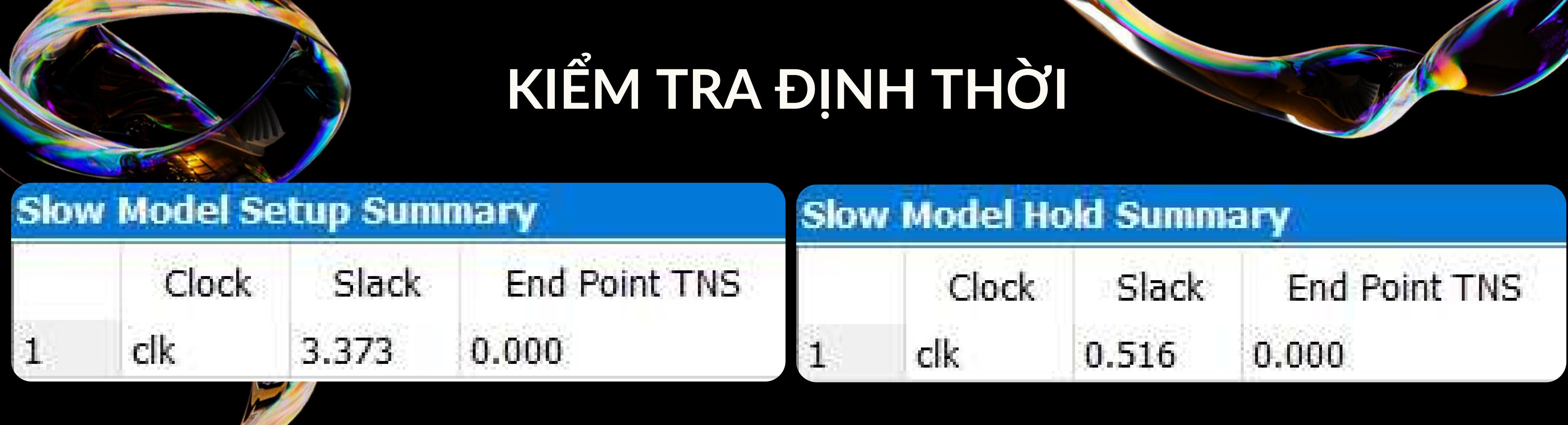
## Slow Model Setup Summary

	Clock	Slack	End Point TNS
1	clk	3.373	0.000

## Slow Model Hold Summary

	Clock	Slack	End Point TNS
1	clk	0.516	0.000

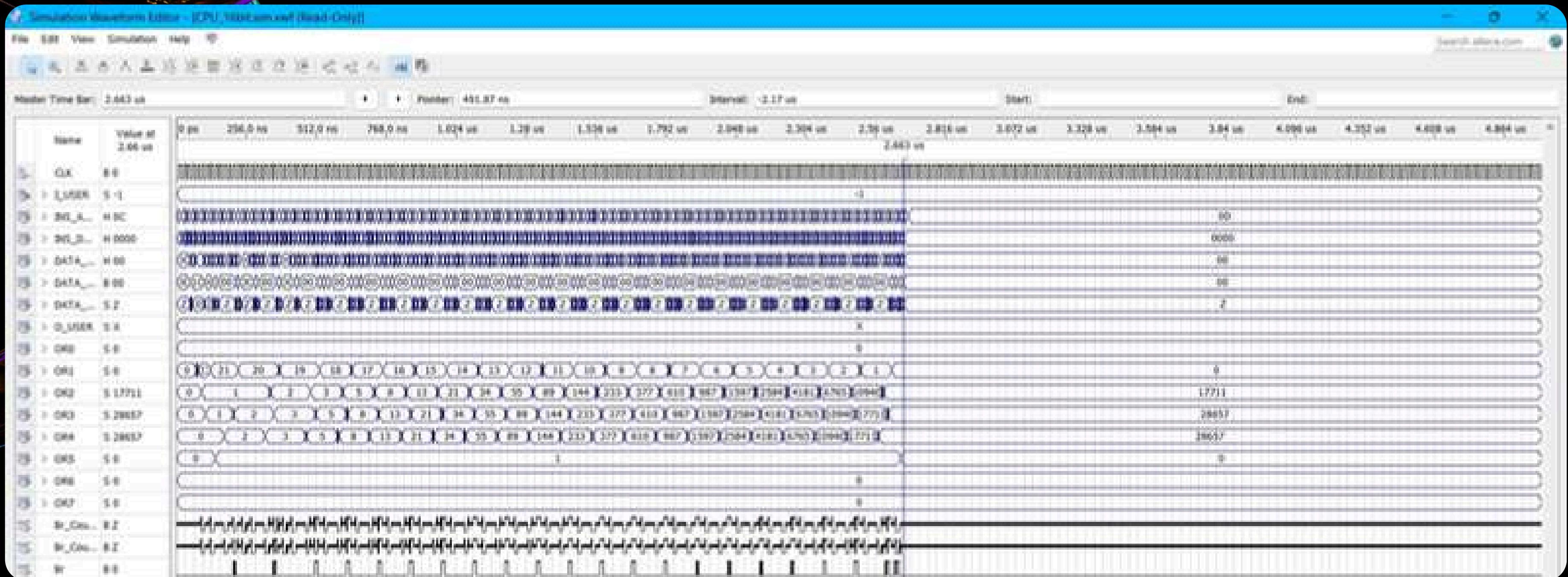
# KIỂM TRA ĐỊNH THỜI



Slow Model Setup: 'clk'								
Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay	
1	CPU_16bit:inst MUX_Sx1_16bit:inst17 DFF_16bit:inst1 inst15	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.662	
2	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.662	
3	3.615	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.418	
4	3.615	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.418
5	3.641	CPU_16bit:inst MEM_WB:inst7 DFF_16bit:inst1 inst15	CPU_16bit:inst MUX_2x1_16bit:inst1 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.393
6	3.641	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst1 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.393
7	3.656	CPU_16bit:inst MEM_WB:inst7 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_3x1_16bit:inst1 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.379
8	3.755	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.008	16.289
9	3.804	CPU_16bit:inst MEM_WB:inst7 DFF_16bit:inst15	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.201
10	3.898	CPU_16bit:inst MEM_WB:inst7 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.135
11	3.924	CPU_16bit:inst MEM_WB:inst7 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst1 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.110
12	3.941	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst1 inst15	CPU_16bit:inst MUX_Sx1_16bit:inst1 MUX_2x1:inst1 inst3~1_OTERM136	clk	clk	20.000	-0.003	16.092
13	3.941	CPU_16bit:inst MEM_WB:inst7 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~1_OTERM136	clk	clk	20.000	-0.003	16.092
14	3.945	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst14	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.003	16.094
15	3.997	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.006	16.045
16	4.023	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst MUX_2x1_16bit:inst1 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.007	16.020
17	4.044	CPU_16bit:inst EX_MEM:inst16 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.000	15.992
18	4.076	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst15	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	15.957
19	4.102	CPU_16bit:inst MEM_WB:inst7 DFF_16bit:inst15	CPU_16bit:inst MUX_2x1_16bit:inst1 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	15.932
20	4.102	CPU_16bit:inst MEM_WB:inst7 DFF_2bit:inst13	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.115	15.919
21	4.104	CPU_16bit:inst RF_2O_8x16:inst1 RFC_2O_4bit:inst1 RFC_2O_20:inst1 inst84	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~1_OTERM126	clk	clk	10.000	0.008	5.940
22	4.110	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst2 inst15	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	15.925
23	4.123	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst1 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX_2x1:inst3 MUX_2x1:inst2 inst~1_OTERM664	clk	clk	20.000	-0.002	15.915
24	4.123	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX_2x1:inst3 MUX_2x1:inst2 inst~1_OTERM664	clk	clk	20.000	-0.002	15.915
25	4.124	CPU_16bit:inst Forwarding_Unit:inst20 inst22~2_OTERM124	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.003	15.909
26	4.131	CPU_16bit:inst RF_2O_8x16:inst1 RFC_2O_4bit:inst1 RFC_2O_20:inst3 inst4	CPU_16bit:inst MUX_Sx1_16bit:inst18 M_X_2x1:inst2 inst3~3_OTERM502	clk	clk	10.000	0.002	5.907
27	4.166	CPU_16bit:inst Forwarding_Unit:inst20 inst1~1_OTERM122	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.002	15.872
28	4.187	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst14	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.001	15.850
29	4.203	CPU_16bit:inst MUX_2x1_16bit:inst11 M_X_2x1:inst1 inst2~0_OTERM196	CPU_16bit:inst MUX_Sx1_16bit:inst18 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.002	15.831
30	4.204	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst1 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX_2x1:inst3 MUX_2x1:inst1 inst3~0_OTERM666	clk	clk	20.000	-0.001	15.833
31	4.204	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX_2x1:inst3 MUX_2x1:inst1 inst3~0_OTERM666	clk	clk	20.000	-0.001	15.833
32	4.205	CPU_16bit:inst MEM_WB:inst7 DFF_16bit:inst1 inst15	CPU_16bit:inst Forwarding_Unit:inst20 inst2~3_OTERM8202	clk	clk	20.000	-0.001	15.832
33	4.205	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst2~3_OTERM8202	clk	clk	20.000	-0.001	15.832

Slow Model Hold: 'clk'							
Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	0.516	CPU_16bit:inst D_REG:inst14 DFF_3bit:inst2 inst2	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst2 inst2	clk	clk	0.000	0.000
2	0.516	CPU_16bit:inst D_REG:inst14 inst18	CPU_16bit:inst REG_EX:inst40 inst5 inst12	clk	clk	0.000	0.000
3	0.516	CPU_16bit:inst D_REG:inst14 DFF_16bit:inst4 inst12	CPU_16bit:inst EX_MEMORY:inst16 DFF_3bit:inst4 inst12	clk	clk	0.000	0.000
4	0.517	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst4 inst12	CPU_16bit:inst REG_EX:inst40 inst4 inst12	clk	clk	0.000	0.000
5	0.518	CPU_16bit:inst D_REG:inst40 inst4 inst12	CPU_16bit:inst REG_EX:inst40 inst5 inst12	clk	clk	0.000	0.000
6	0.518	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst4 inst12	CPU_16bit:inst REG_EX:inst40 inst4 inst12	clk	clk	0.000	0.000
7	0.518	CPU_16bit:inst REG_EX:inst40 inst4 inst12	CPU_16bit:inst REG_EX:inst40 inst4 inst12	clk	clk	0.000	0.000
8	0.519	CPU_16bit:inst D_REG:inst14 inst5 inst12	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst2 inst2	clk	clk	0.000	0.000
9	0.519	CPU_16bit:inst D_REG:inst14 DFF_16bit:inst18 inst4	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst4	clk	clk	0.000	0.000
10	0.520	CPU_16bit:inst D_REG:inst14 DFF_16bit:inst10 inst10	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst10 inst10	clk	clk	0.000	0.000
11	0.520	CPU_16bit:inst EX_MEMORY:inst16 DFF_16bit:inst25 inst15 inst13	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst15 inst13	clk	clk	0.000	0.000
12	0.523	CPU_16bit:inst REG_EX:inst14 DFF_16bit:inst19 inst5	CPU_16bit:inst REG_EX:inst14 DFF_16bit:inst19 inst5	clk	clk	0.000	0.000
13	0.527	CPU_16bit:inst EX_MEMORY:inst16 DFF_16bit:inst19 inst5	CPU_16bit:inst REG_EX:inst14 DFF_16bit:inst19 inst5	clk	clk	0.000	0.000
14	0.530	CPU_16bit:inst D_REG:inst14 DFF_16bit:inst18 inst6	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst6	clk	clk	0.000	0.000
15	0.531	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst5 inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst5 inst6	clk	clk	0.000	0.000
16	0.539	CPU_16bit:inst EX_MEMORY:inst16 DFF_16bit:inst25 inst6 inst6	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst6 inst6	clk	clk	0.000	0.000
17	0.545	CPU_16bit:inst D_REG:inst14 DFF_3bit:inst6 inst2	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst6 inst2	clk	clk	0.000	0.000
18	0.549	CPU_16bit:inst EX_MEMORY:inst16 DFF_16bit:inst25 inst10 inst10	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst10 inst10	clk	clk	0.000	0.000

# KIỂM TRA ĐỊNH THỜI



# KIỂM TRA CÔNG SUẤT

## PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Wed Dec 17 18:02:14 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	CPU_16bit
Top-level Entity Name	Datapath
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	217.47 mW
Core Dynamic Thermal Power Dissipation	23.72 mW
Core Static Thermal Power Dissipation	80.28 mW
I/O Thermal Power Dissipation	113.47 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

# TỔNG HỢP KẾT QUẢ

Tiêu chí	CPU 16-bit	Nios II/s (Standard)
Tài nguyên	<ul style="list-style-type: none"> <li>- Logic elements: <b>3,128</b></li> <li>- Registers: <b>541</b></li> </ul>	LEs: $\approx 1,400 - 1,800$
Tần số tối đa	<ul style="list-style-type: none"> <li>- <math>F_{\max} \approx 60.14</math> MHz</li> <li>- <math>T_{\min} \approx 16.63</math> ns</li> </ul>	$\approx 80 - 100$ MHz (trên Cyclone II)
Định thời	<ul style="list-style-type: none"> <li>- Setup slack: <b>3.373</b> ns</li> <li>- Hold slack: <b>0.516</b> ns</li> </ul>	...
Công suất	<ul style="list-style-type: none"> <li>- Total Thermal Power Dissipation: <b>217.47</b> mW</li> <li>- Core Dynamic Thermal Power Dissipation: <b>23.72</b> mW</li> <li>- Core Static Thermal Power Dissipation: <b>80.28</b> mW</li> <li>- I/O Thermal Power Dissipation: <b>113.47</b> mW</li> </ul>	<ul style="list-style-type: none"> <li>- Core Dynamic Thermal Power Dissipation: <b>10 - 15</b> mW</li> </ul>
Hiệu năng (Testbench: Tìm số Fibonacci thứ 23)	<ul style="list-style-type: none"> <li>- Thời gian thực thi: <math>\approx 2.663</math> <math>\mu</math>s</li> <li>- Testbench:</li> <li>    li r1, 23</li> <li>    li r2, 1</li> <li>    li r3, 1</li> <li>    subi r1, r1, 2</li> <li>    slt r5, r0, r1</li> <li>    beq r5, r0, 11</li> <li>    add r4, r3, r2</li> <li>    mov r2, r3</li> <li>    mov r3, r4</li> <li>    dec r1, r1</li> <li>    jmp 4</li> </ul>	<ul style="list-style-type: none"> <li>- Thời gian thực thi: <math>\approx 2.5 - 3.5</math> <math>\mu</math>s</li> </ul>

Nhóm 1 - CE118.Q11.VMTN

# THANK YOU

for your time and attention

**GitHub Project:**

