

**ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN TỰ ĐỘNG HÓA**

-----



**CUỐI KỲ THIẾT KẾ SỐ**

**TÊN ĐỀ TÀI**

**NGHIÊN CỨU RISC 8 BIT MÔ PHỎNG LÊN VIVADO**

**SVTH: LÊ THẠCH ANH TÀI**  
**MSHV: 2002210206**

**TP. Hồ Chí Minh, .../.../20...**



# NỘI DUNG CHÍNH

---

**I. KIẾN TRÚC VONEUMAN**

**II. CHIP A12**

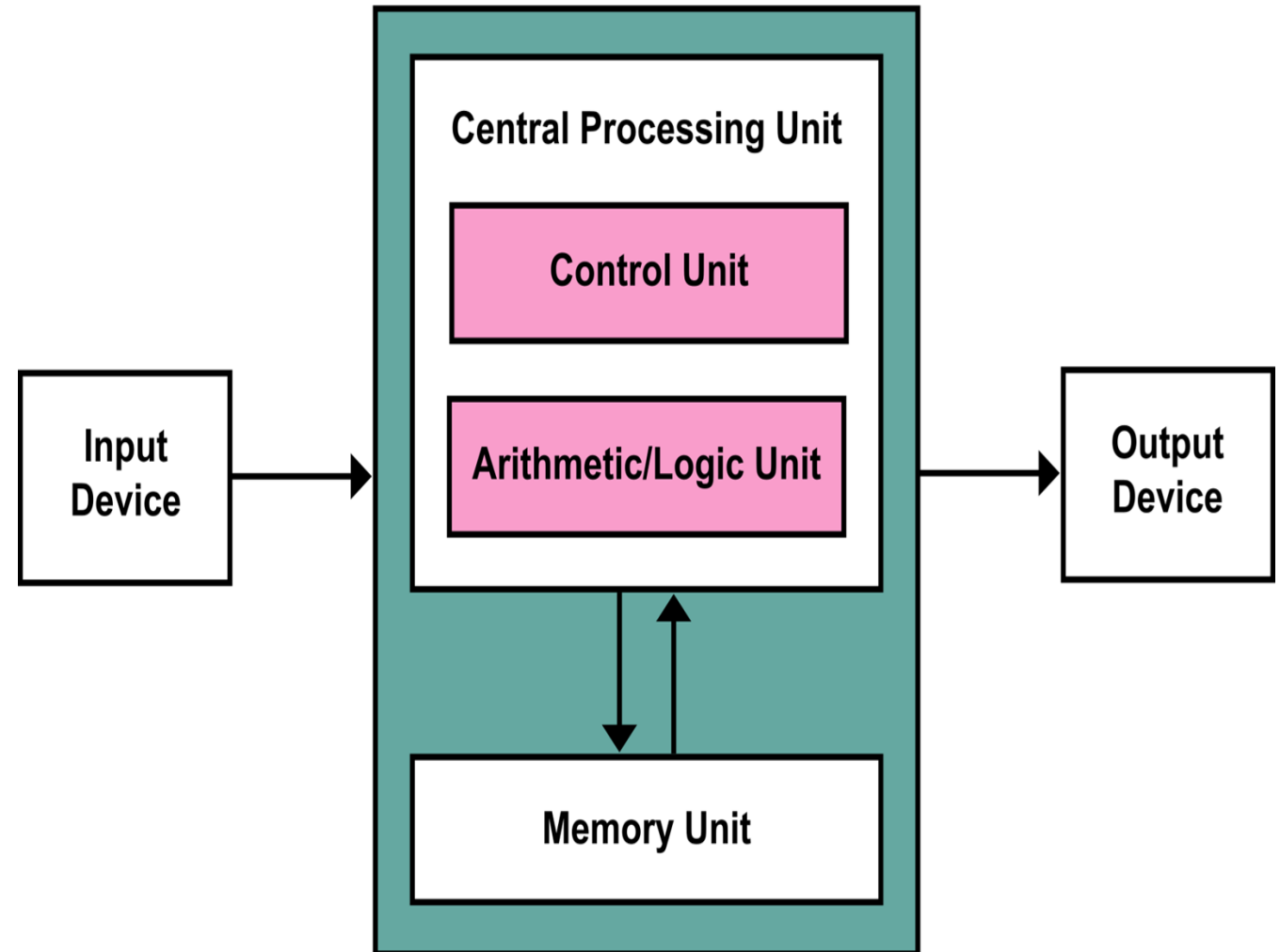
**III. 3 KHỐI CHỨC NĂNG**

**IV. KẾT QUẢ THỰC NGHIỆM**

# I. KIẾN TRÚC VONEUMAN

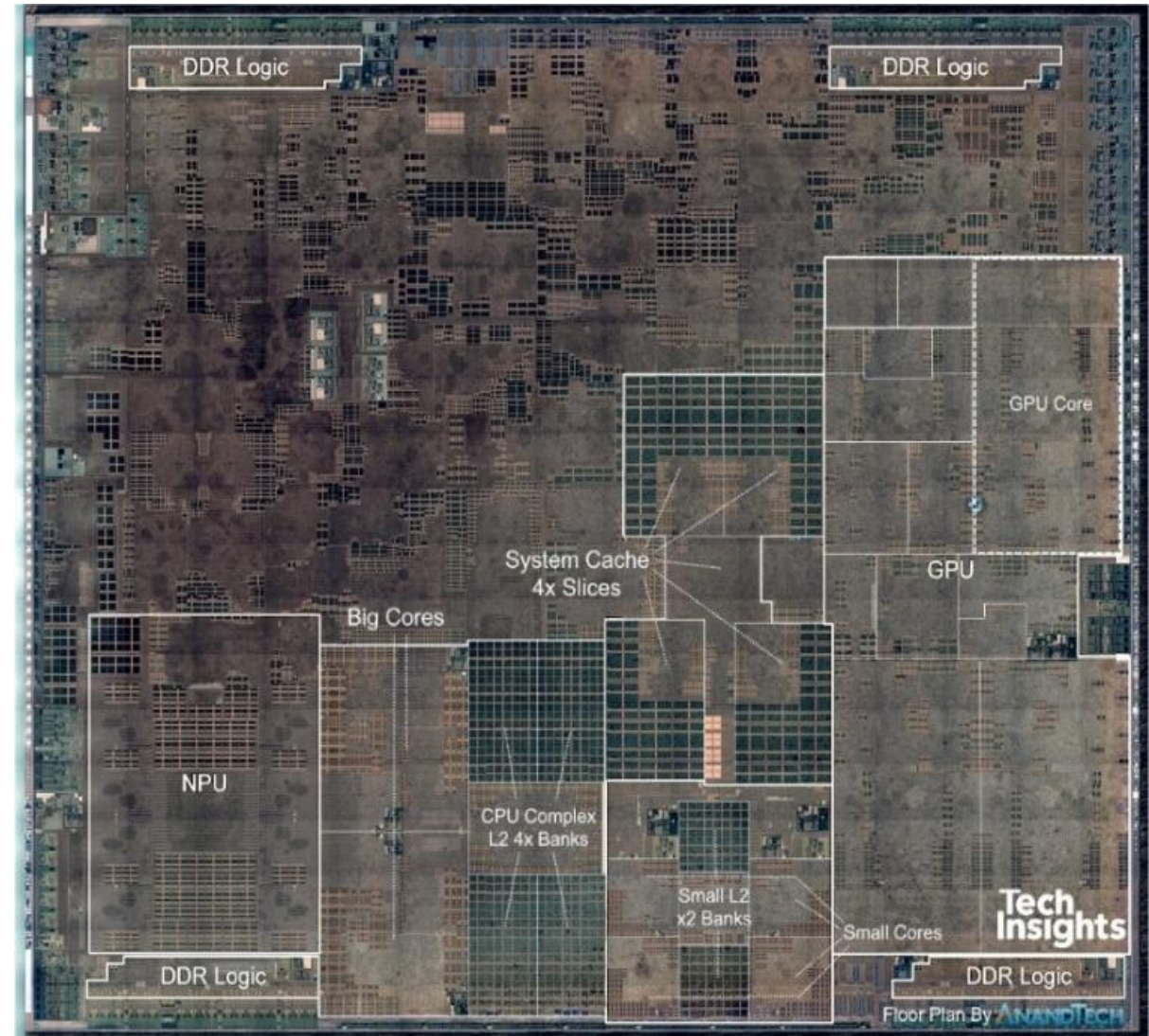
## Kiến trúc Von Neumann

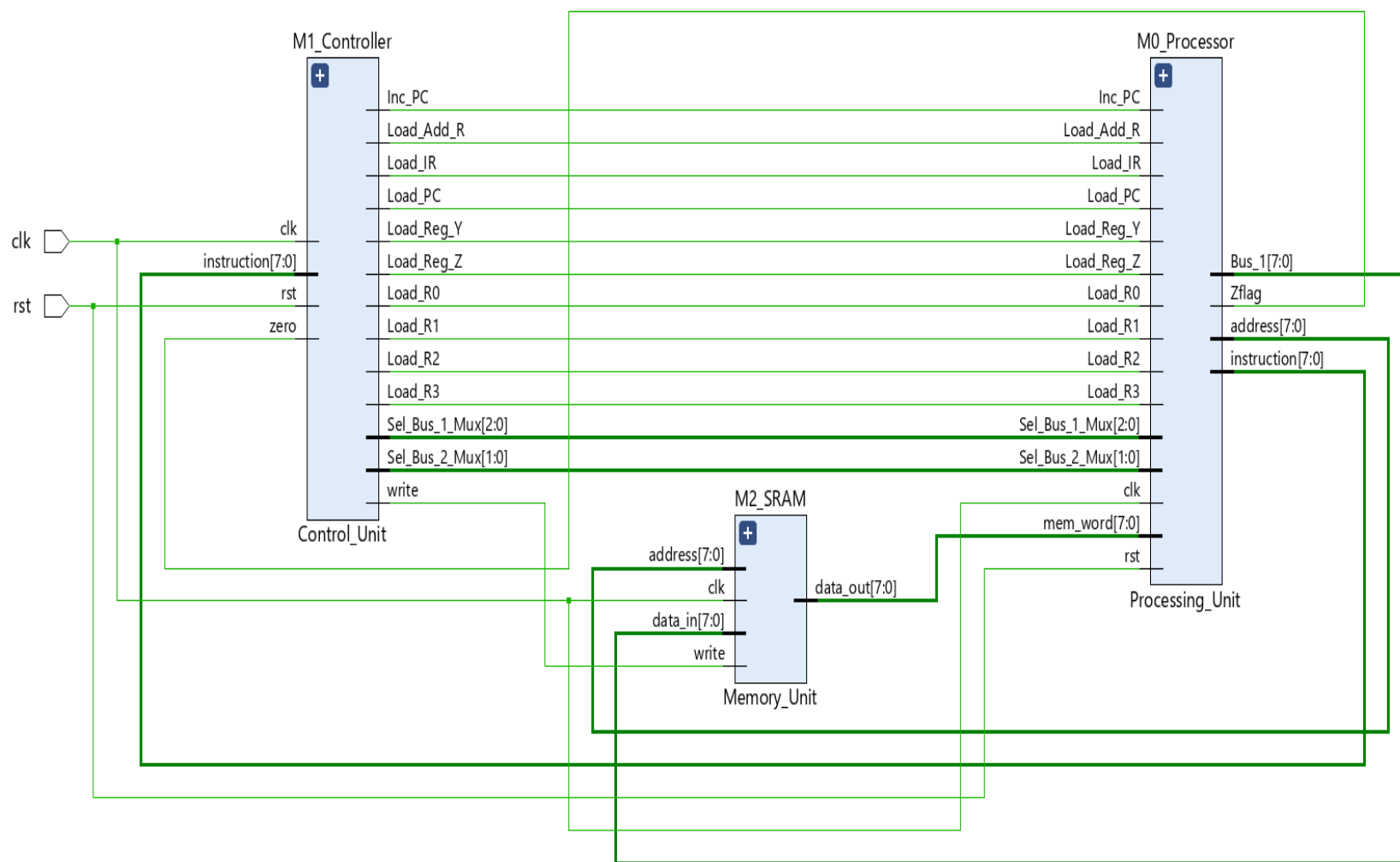
Hiện nay trên thời giới bất kỳ CPU nào bên trong đều là kiến trúc Von Neumann. Gồm 3 khối Control Unit, Processor, Memory Unit



## II. CHIP A12

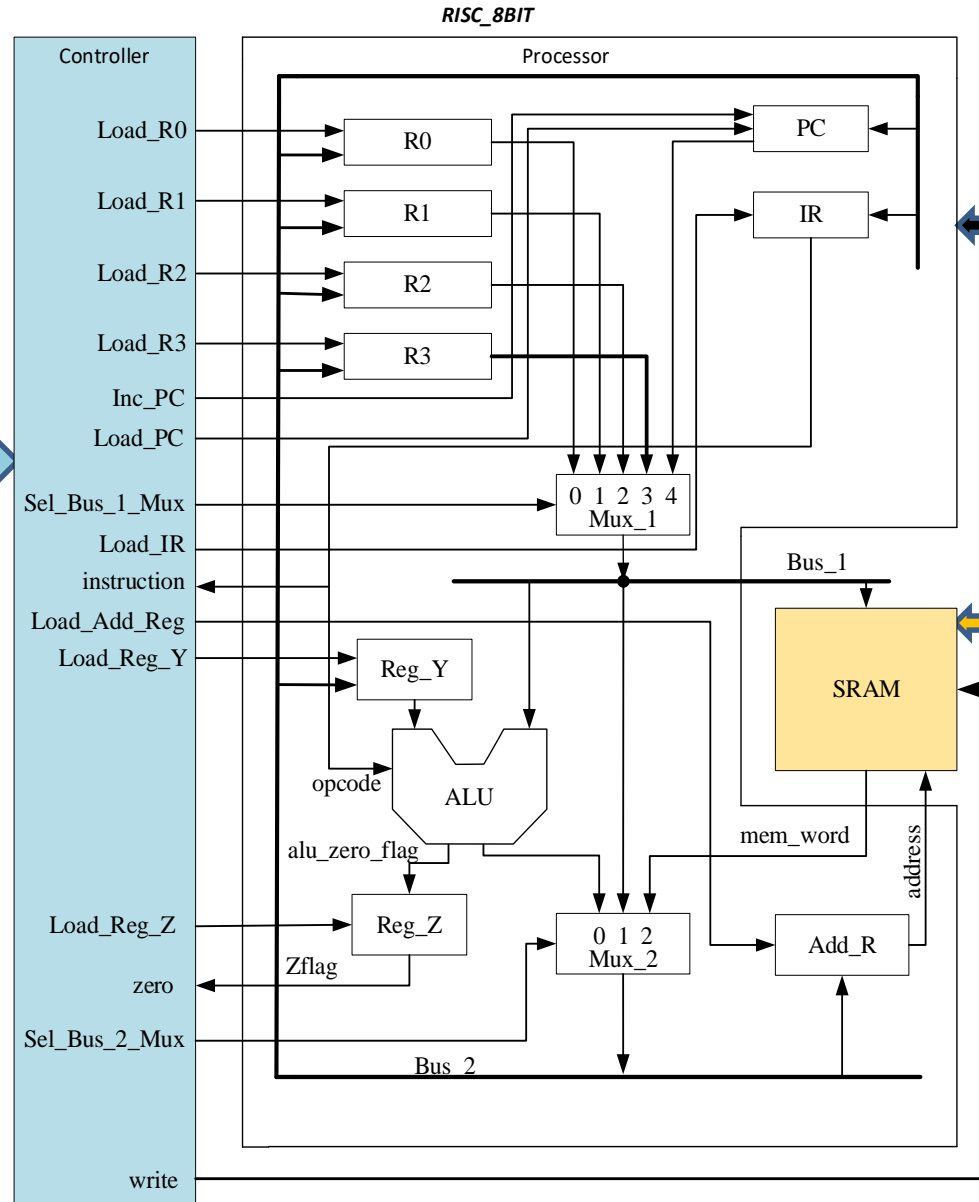
- CPU và bộ nhớ chip A12
  - Chip A12 trên iphone XS ta thấy có CPU và bộ nhớ Cache.
  - CPU chứa 2 khối điều khiển và khối xử lý
  - Bộ nhớ Cache cấu tạo từ các SRAM





# III. 3 KHỐI CHỨC NĂNG

Bộ điều khiển



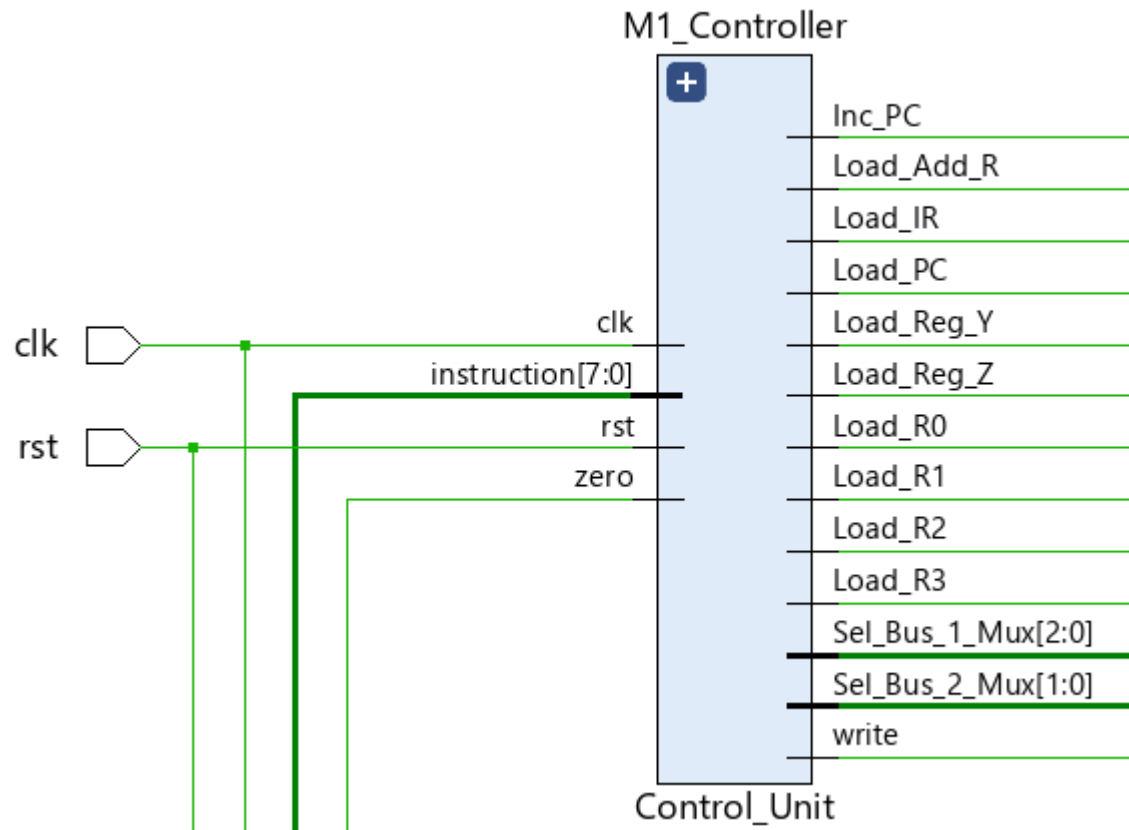
Bộ xử lý



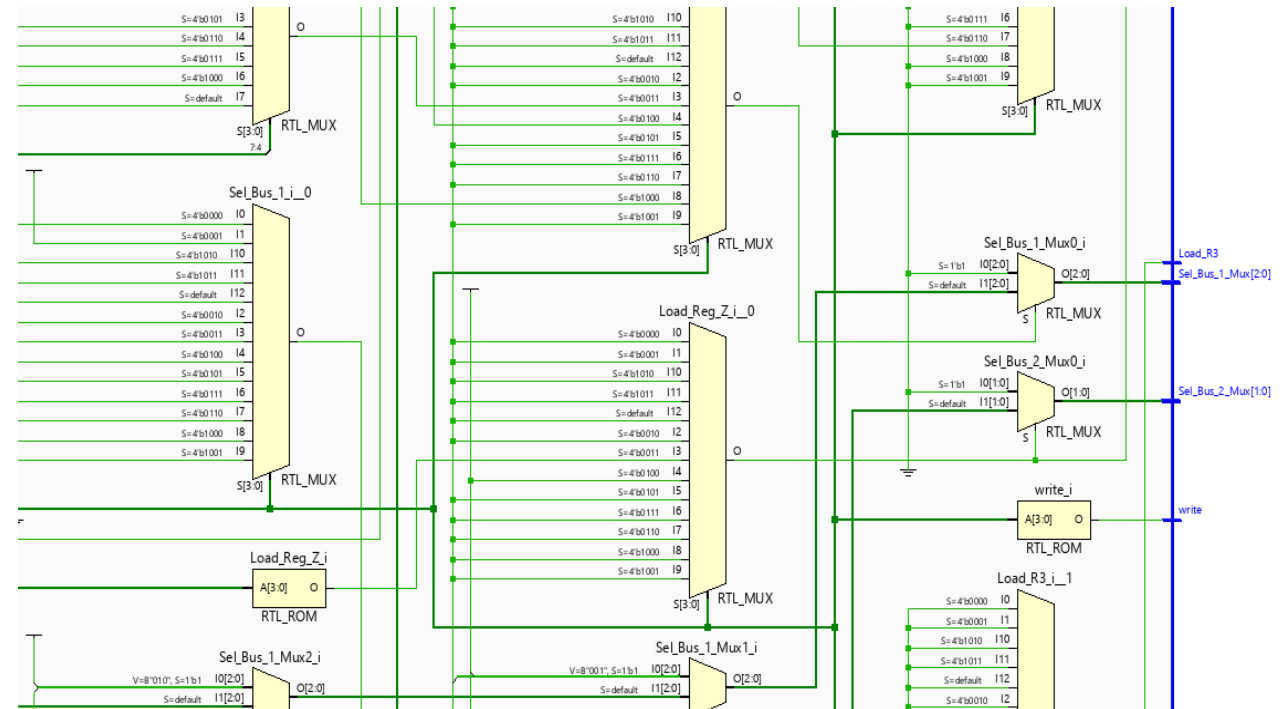
Bộ nhớ



# III. BỘ ĐIỀU KHIỂN

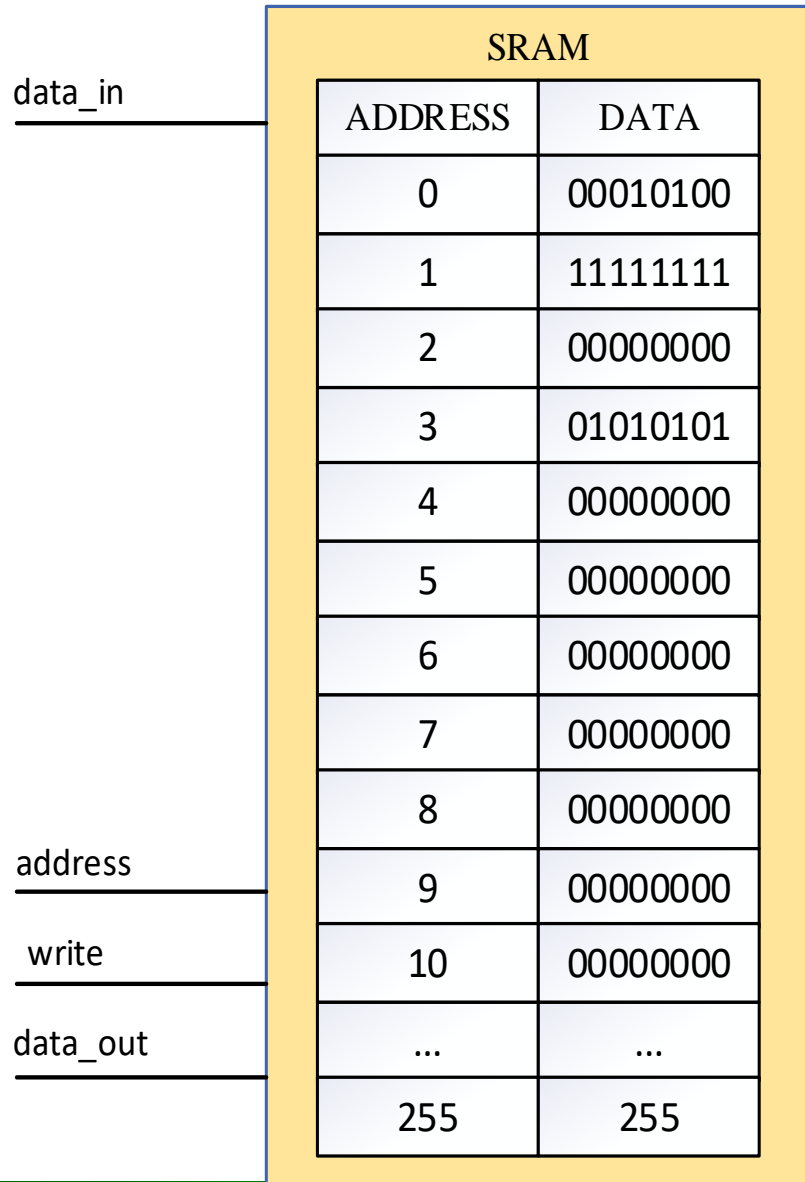


## • Bộ điều khiển



### III. BỘ NHỚ TĨNH

- **Bộ nhớ nhanh**
  - Bao gồm 255 địa chỉ, mỗi địa chỉ chứa 8bit.





# III. CÂU LỆNH 8 BIT

## • RISC 8bit

### Lệnh ngắn

Opcode				Nguồn		Đích	
0	0	0	1	0	0	1	0

### Lệnh dài

Opcode				Nguồn		Đích	
0	1	0	1	1	0	-	-
Địa chỉ							
0	0	0	1	1	1	0	1

Lệnh	Lệnh 8 bit			Hành động
	opcode	nguồn	đích	
NOP	0000	--	--	không làm gì
ADD	0001	nguồn	đích	đích $\leq$ nguồn + đích
SUB	0010	nguồn	đích	đích $\leq$ đích – nguồn
AND	0011	nguồn	đích	đích $\leq$ nguồn & đích
NOT	0100	nguồn	đích	đích $\leq$ ~ nguồn
RD	0101	--	đích	Lấy giá trị từ ô nhớ trong địa chỉ lưu vào đích
WR	0110	nguồn	--	Đọc giá trị từ ô nhớ trong địa chỉ lưu vào nguồn
BR	0111	--	--	Đọc giá trị từ ô nhớ trong địa chỉ lưu vào PC
BRZ	1000	--	--	Đọc giá trị từ ô nhớ trong địa chỉ lưu vào PC
HALT	111	--	--	dừng tất cả lại đến khi reset

# IV. LỆNH NẠP

Các giá trị trong ngoặc vuông là các giá trị địa chỉ

SRAM[0] = 8'b0101\_00\_10

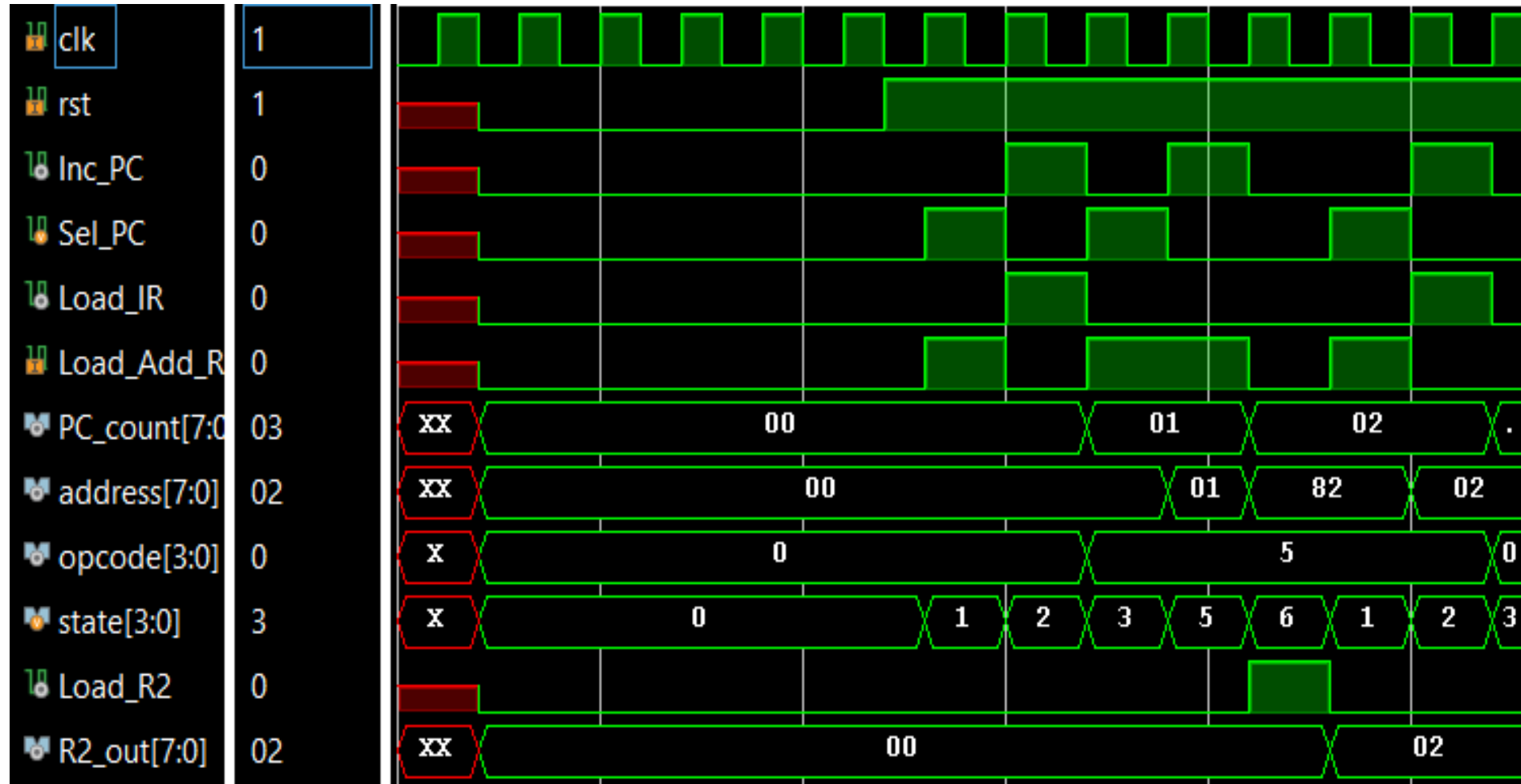
SRAM [1] = 130;

SRAM [130] = 2;

Nạp giá trị 2 ở ô nhớ số 130

vào thanh ghi đa năng R2

tương ứng với 2 bit cuối 10 ở địa chỉ 0



SRAM[0] = 0000\_00\_00

SRAM[1] = 0101\_00\_10

SRAM[2] = 130

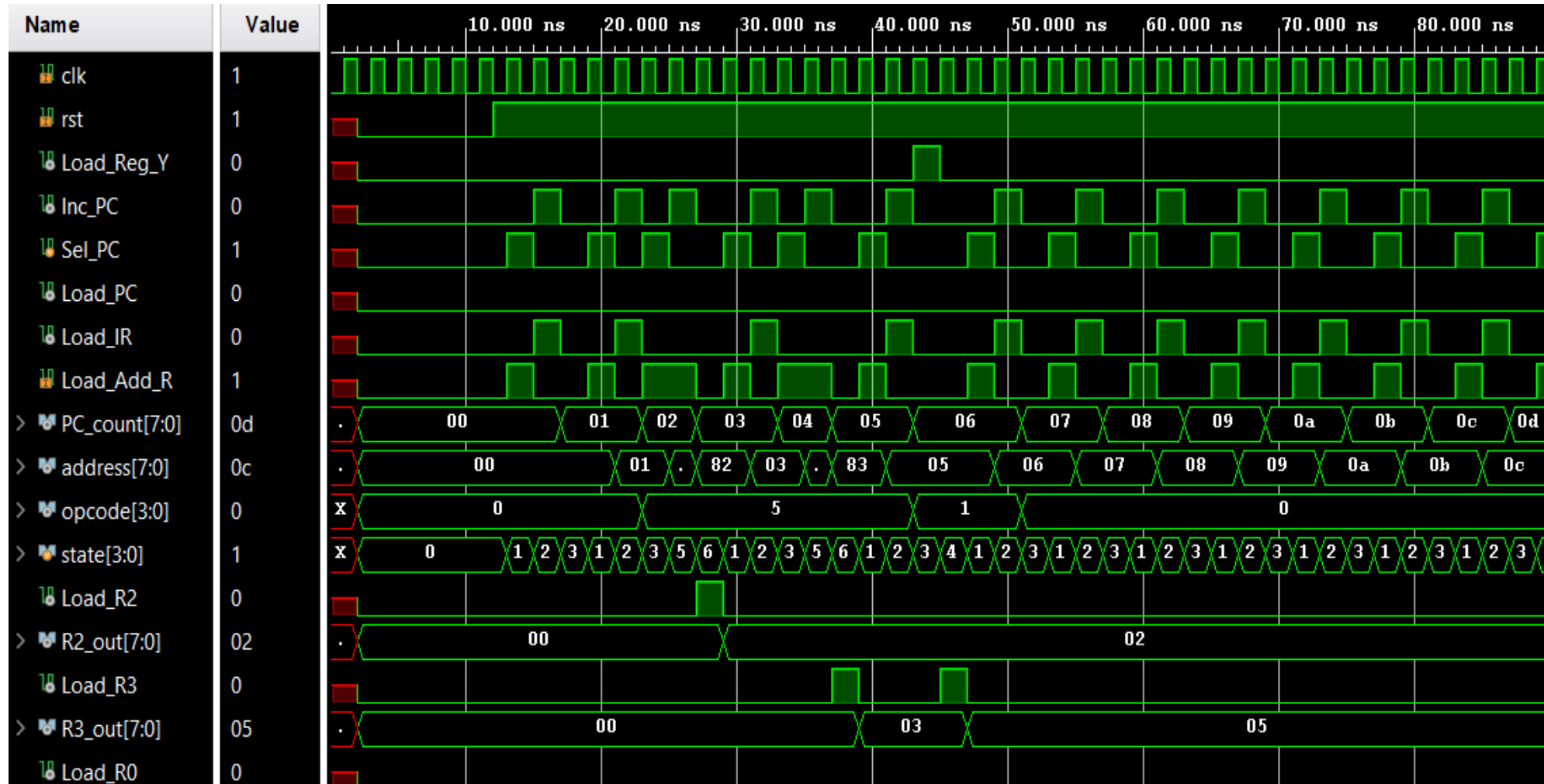
SRAM[130] = 2

SRAM[3] = 0101\_00\_11

SRAM[4] = 131

SRAM[131] = 3

SRAM[5] = 0001\_10\_11



SRAM[0]= 8'b0000\_00\_00

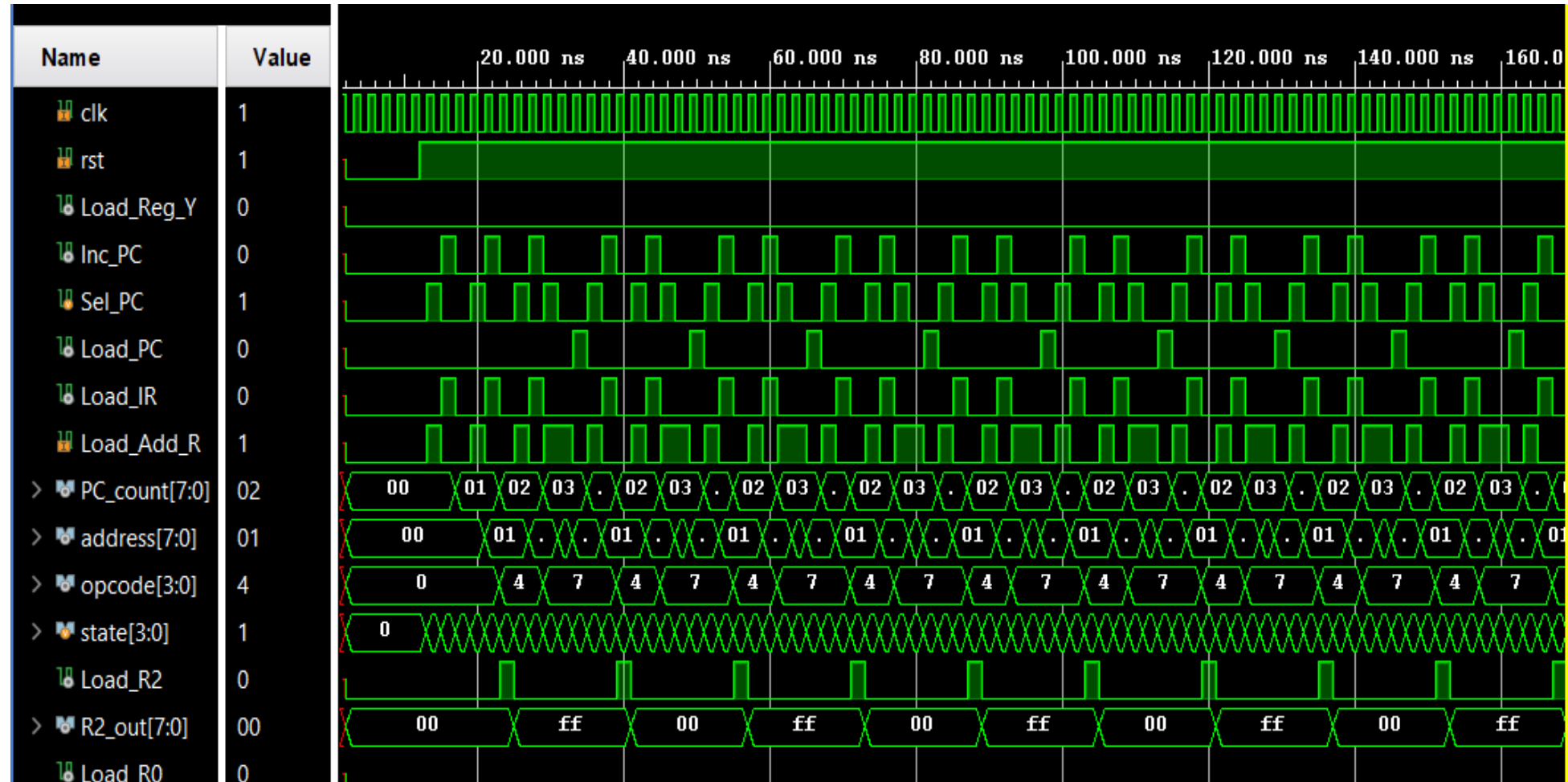
SRAM[1]= 8'b0100\_10\_10

SRAM[3]= 8'b0111\_00\_00

SRAM[4] = 140

SRAM[140] = 1

00000000 =00 ~ 11111111 =ff



SRAM [0] = 8'b0000\_00\_00;

SRAM[1] = 8'b0101\_00\_10;

SRAM[2] = 130;

SRAM[3] = 8'b0101\_00\_11;

SRAM[4] = 131;

SRAM[5] = 8'b0001\_10\_11;

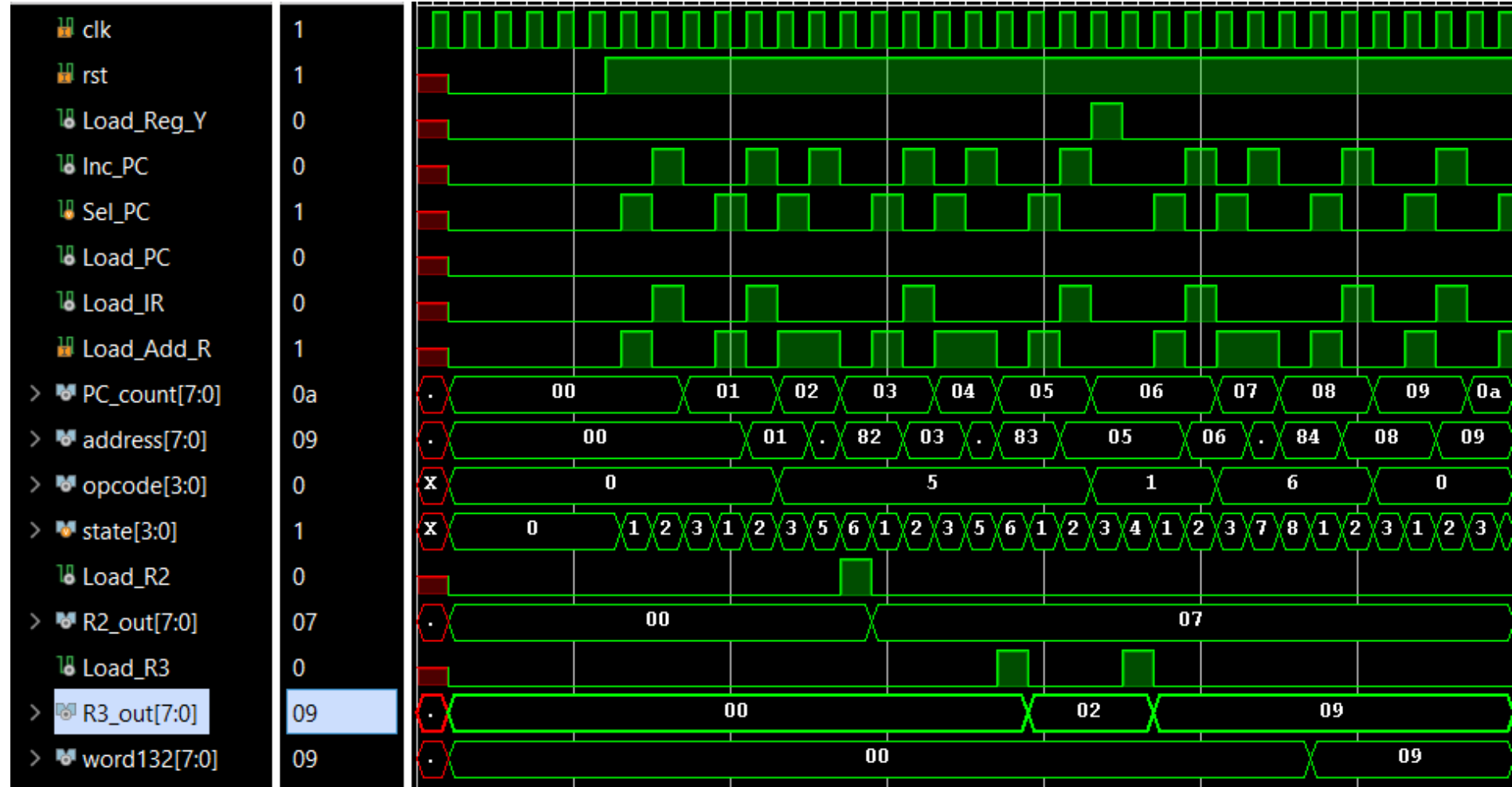
SRAM[6] = 8'b0110\_11\_00;

SRAM[7] = 132;

SRAM[130] = 7;

SRAM[131] = 2;

SRAM[132] = 0;



SRAM [0] = 8'b0000\_00\_00;

SRAM[1] = 8'b0101\_00\_10;

SRAM[2] = 130;

SRAM[3] = 8'b0101\_00\_11;

SRAM[4] = 131;

SRAM[5] = 8'b0001\_10\_11;

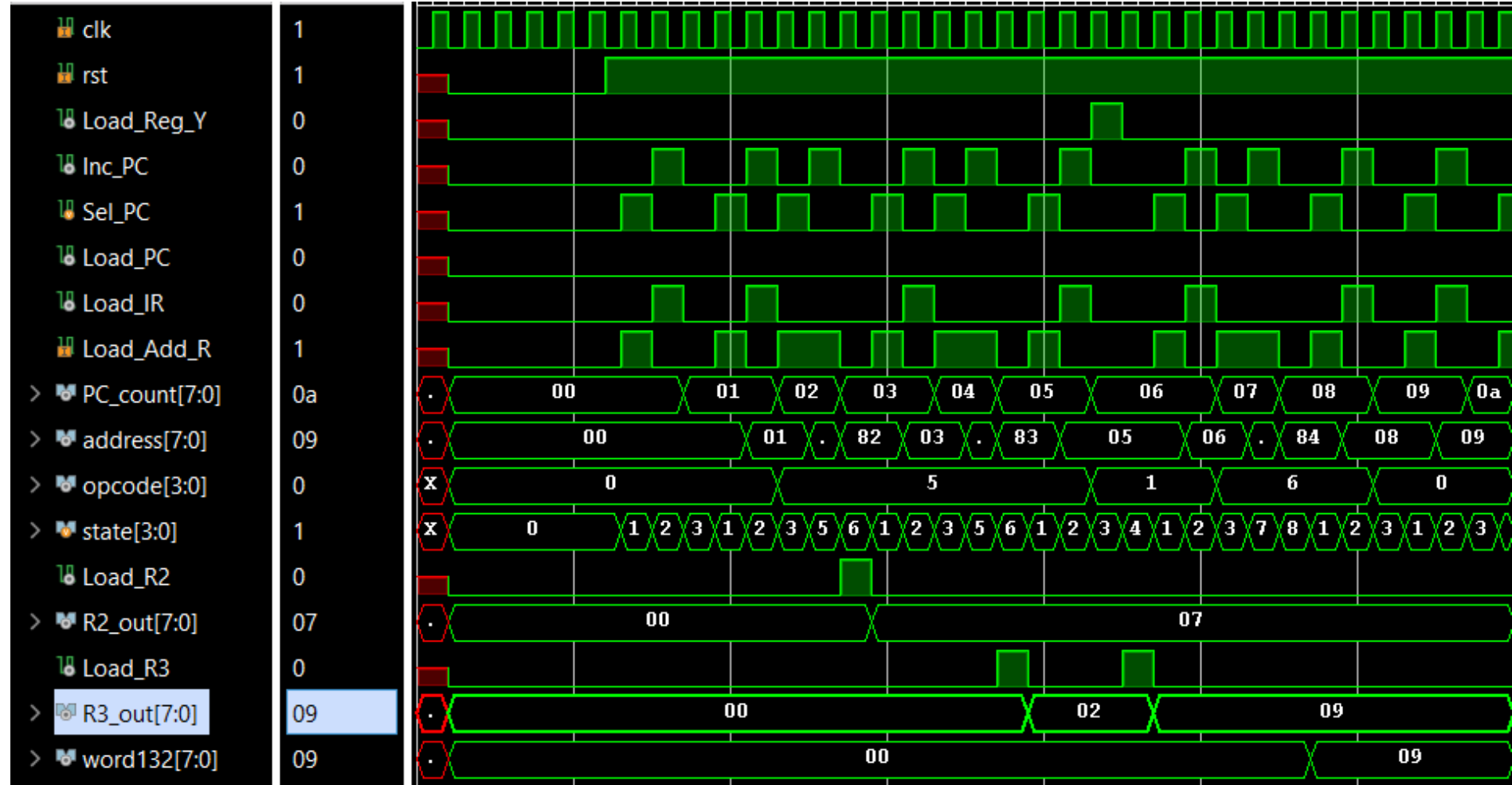
SRAM[6] = 8'b0110\_11\_00;

SRAM[7] = 132;

SRAM[130] = 7;

SRAM[131] = 2;

SRAM[132] = 0;





```
SRAM [140] = 9;           // Recycle
```

TP. HCM, 15/01/2025



**CẢM ƠN CÁC ANH/CHỊ VÀ CÁC BẠN  
ĐÃ XEM**