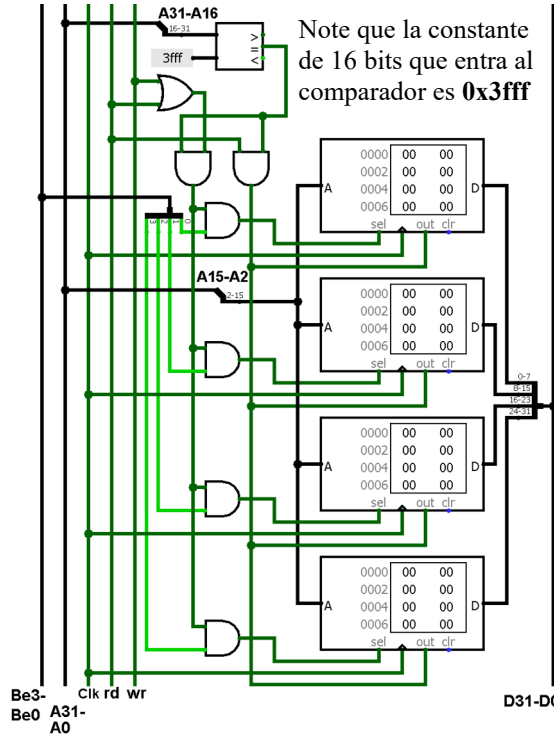


### Pregunta 1

Parte a.- En el circuito de abajo se ha agregado memoria a LRV32IM.

Note que al comparador se conectan las línea A31-A16

Note que a la dirección de los módulos de memoria se conectan las líneas A15-A2



Note que la constante de 16 bits que entra al comparador es 0x3fff

Conteste la siguientes preguntas:

- i. ¿Cuanta memoria se agregó?
- ii. ¿En qué rango de direcciones se agregó?
- iii. ¿Por qué no se conectaron las líneas A1 y A0 a la dirección de los módulos de memoria?
- iv. ¿Qué pasaría si se ignoraran las líneas BE3 a BE0 en el circuito?

Ayuda: 0x40000000 ≡ 1 GB

Parte b.- Si se necesitara agregar 16 MB en el rango de direcciones [2 GB, 2 GB + 16 MB]

- i. ¿Qué líneas del bus de direcciones conectaría al comparador?
- ii. ¿Cuál sería el valor de la constante que conectaría al comparador y de qué tamaño en bits?
- iii. ¿Qué líneas del bus de direcciones conectaría a la dirección en los módulos de memoria?

### Pregunta 2

Parte a.- La función  $f$  de la derecha está programada en assembler Risc-V. Considere que se invoca  $f$  recibiendo en  $a1$  el entero 5 y en  $a0$  la dirección  $d$  de un arreglo de 5 enteros con los valores 3, 8, 1, 7 y 0. La tabla de más abajo muestra la ejecución de la función hasta el 2<sup>do</sup>  $sw$ . **Prosiga llenando la tabla** con la ejecución de  $addi\ a0,a0,4$  hasta que se ejecute la instrucción  $ret$ .

```
f:
    slli a4,a1,2
    add a1,a0,a4
    addi a1,a1,-4
L2:
    lw a4,0(a0)
    lw a3,0(a1)
    sw a3,0(a0)
    sw a4,0(a1)
    addi a0,a0,4
    addi a1,a1,-4
    bltu a0,a1,L2
    ret
```

Instrucción	a0	a1	a3	a4	arreglo d
	d	5			3 8 1 7 0
slli a4,a1,2				20	
add a1,a0,a4		d+20			
addi a1,a1,-4		d+16			
lw a4,0(a0)				3	
lw a3,0(a1)			0		
sw a3,0(a0)					0 8 1 7 0
sw a4,0(a1)					0 8 1 7 <u>3</u>

Parte b.- Traduzca la función  $g$  de más abajo a assembler Risc-V. Optimice el código en assembler para reducir la cantidad de instrucciones.

```
Nodo *g(Nodo *a, int z, int *pres) {
    while ( a!=NULL && a->x!=z ) {
        if (z<a->x)
            a= a->izq;
        else
            a= a->der;
    }
    if (a!=NULL) {
        *pres= a->y;
    }
    return a;
}
```

```
typedef struct nodo {
    int x, y;
    struct nodo *izq, *der;
} Nodo;
```