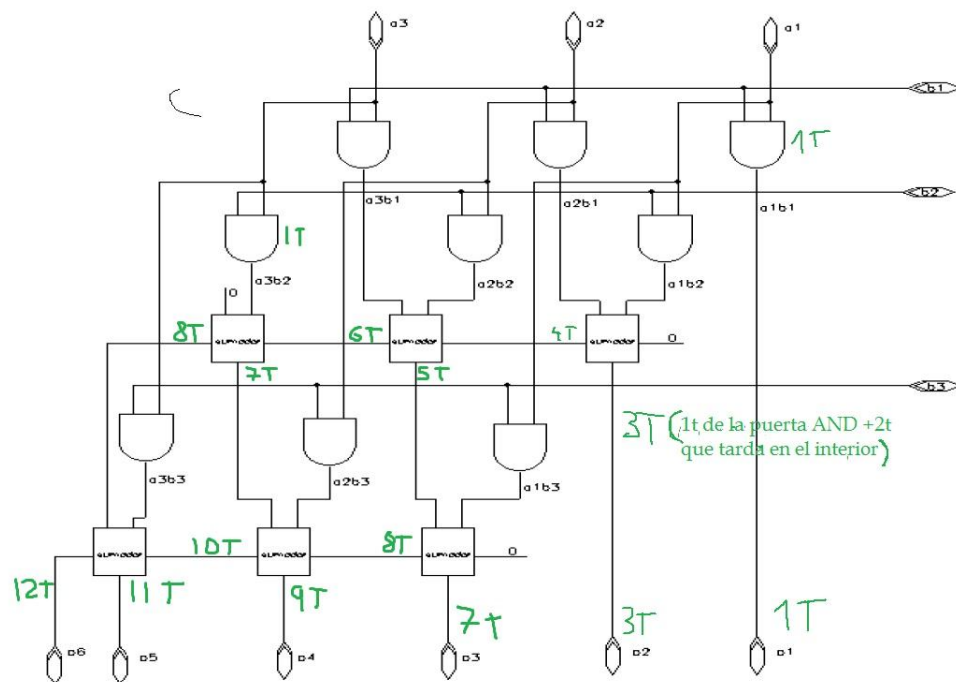


Ejercicio 1

a)



b)

Ejercicio 2

a) y b)

Cada intro entre instrucciones quiere decir que se ejecutan en 1 ciclo. Las señales son las que se activan y necesitamos en cada ciclo las demás no importan en ese momento, pueden estar activas o no.

Tipo R (add, sub, and, or y slt)

RI <- M[PC]	MemRead=1, IRWrite=1, IorD=0
PC <- PC + 4	ALUSrcA=0, ALUSrcB=01, PCWrite=1, ALUOp=00, PCSrc=00
A <- BR[rs]	ALUSrcA=0,
B <- BR[rt]	ALUSrcB=11
ALUoutput <- PC + ext_signo(desplazamiento << 2)	ALUOp=00
ALUoutput <- A op B	ALUSrcA=1, ALUOp=10, ALUSrcB=00
BR[rd] <- ALUoutput	RegDst=1, RegWrite=1, MemtoReg=0

Tipo I (lw, sw, beq)

Lw

RI <- M[PC]	MemRead=1, IRWrite=1, lorD=0
PC <- PC + 4	ALUSrcA=0, ALUSrcB=01, PCWrite=1, ALUOp=00, PCSource=00
A <- BR[rs]	ALUSrcA=0,
B <- BR[rt]	ALUSrcB=11
ALUoutput <- PC + ext_signo(desplazamiento << 2)	ALUOp=00
ALUoutput <- A + ext_signo(desplazamiento)	ALUSrcA=1, ALUSrcB=10, ALUOp=00
MDR <- M[ALUoutput]	MemRead=1, lorD=1
BR[rt] <- MDR	RegWrite=1, MemtoReg=1

Sw

RI <- M[PC]	MemRead=1, IRWrite=1, lorD=0
PC <- PC + 4	ALUSrcA=0, ALUSrcB=01, PCWrite=1, ALUOp=00, PCSource=00
A <- BR[rs]	ALUSrcA=0,
B <- BR[rt]	ALUSrcB=11
ALUoutput <- PC + ext_signo(desplazamiento << 2)	ALUOp=00
ALUoutput <- A + ext_signo(desplazamiento)	ALUSrcA=1, ALUSrcB=10, ALUOp=00
M[ALUoutput] <- B	MemWrite=1,lorD=1

Beq

RI <- M[PC]	MemRead=1, IRWrite=1, lorD=0
PC <- PC + 4	ALUSrcA=0, ALUSrcB=01, PCWrite=1, ALUOp=00, PCSource=00
A <- BR[rs]	ALUSrcA=0,
B <- BR[rt]	ALUSrcB=11
ALUoutput <- PC + ext_signo(desplazamiento << 2)	ALUOp=00
Si (A==B) entonces	ALUSrcA=1, ALUOp=01, PCWriteCond=1, PCSource=01, ALUSrcB=00
PC <- ALUoutput	

Tipo J (j)

RI <- M[PC]	MemRead=1, IRWrite=1, lrd=0
PC <- PC + 4	ALUSrcA=0, ALUSrcB=01, PCWrite=1, ALUOp=00, PCSource=00
A <- BR[rs]	ALUSrcA=0,
B <- BR[rt]	ALUSrcB=11
ALUoutput <- PC + ext_signo(desplazamiento << 2)	ALUOp=00
PC <- PC[RI[31-28]] & (RI[25-0] << 2)	PCWrite=1, PCSource=10

c)

- La señal de control Write_A se puede eliminar.
- La señal de control Write_B se puede eliminar.
- La señal de control Write_ALUout se puede eliminar.

Una señal de control nos permite elegir si una unidad funcional se activa o no (muestra su contenido) o para seleccionar un dato de otro (p. ejemplo ALUSrcB con el MUX).

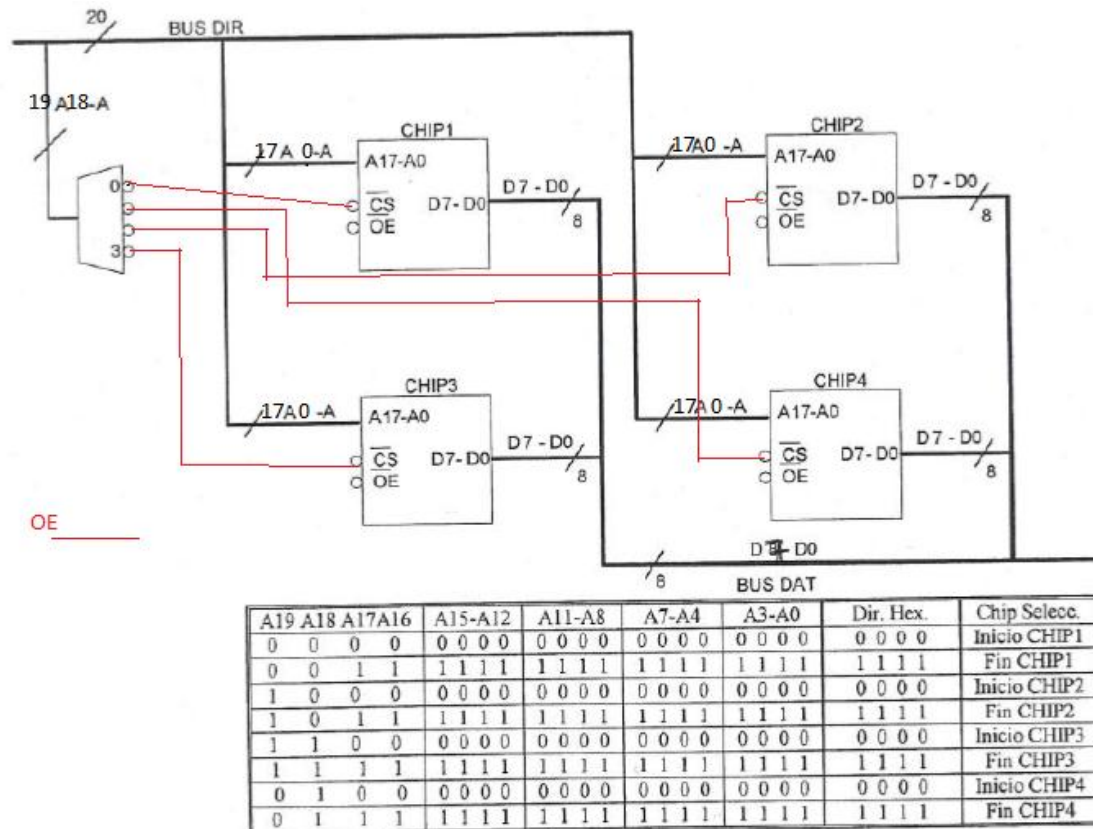
Las señales Write_A y Write_ALUout de los registros temporales A y ALUout se pueden eliminar ya que siempre se va a escribir en ellos, no tiene sentido poner una señal para controlar si se escribe o no, si siempre se va a escribir.

Por otro lado, la señal Write_B no se puede eliminar porque como el banco de registros solo puede leer un registro hay que controlar si lo guardamos en el temporal A o B. Ya que si no lo controlamos al leer el primer registro fuente se guardaría en A y B, luego leemos el segundo registro fuente y se vuelve a guardar en A y B, por lo tanto tendríamos el dato de uno de los registros fuentes en los dos registros temporales A y B.

En conclusión, necesitamos una señal para controlar cuando escribir en A y B, que puede ser o Write_A o Write_B. Una de las dos tienen que existir. Y la señal Write_ALUout se puede eliminar.

Ejercicio 3

- a) La línea OE iría conectada a cada entrada OE de los chips. No corresponde a ningún bit de la dirección (Ax).
Si OE está activa entonces el chip volcará la información al bus de datos.



- b) La memoria tiene un tamaño de 2GByte (2Gx8) y está estructurada en palabras de 32 bits, por lo que hay que convertir 2Gx8 a "algo x 32".

Como 32 son 4 veces 8 ($32/8=4$) tendremos que dividir 2G entre 4.

Pasandolo a megas: 2G = 2048 Mb.

Entonces $2048/4 = 512\text{Mx}32$.

Bus de direcciones (m) = $512\text{M} = 2^9 \times 2^{20} = 2^{29} \Rightarrow 29$ bits

Bus de datos (n) = 32 bits