

[illegible]

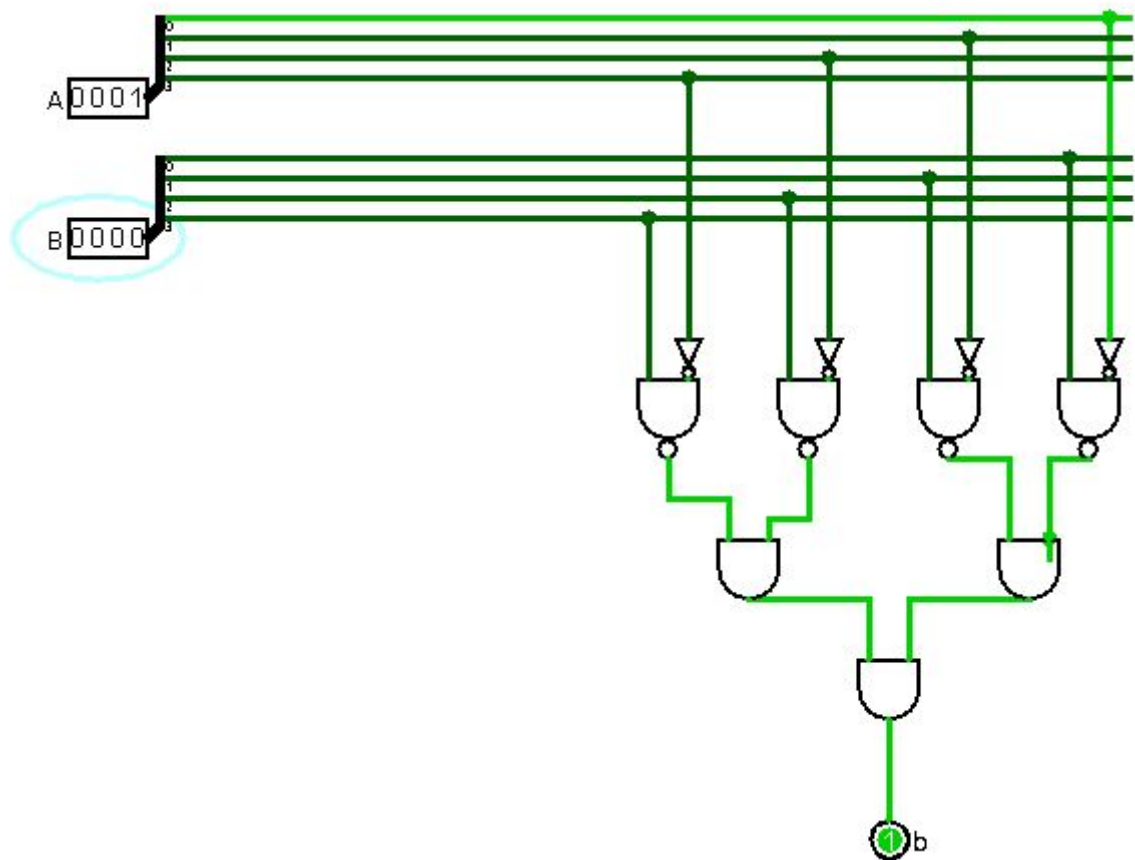
- c) A = parte alta en r13, parte baja en r12
B = parte alta en r15, parte baja en r14

en r10 guardo la parte baja de la respuesta y en r11 la alta
en r9 guardo el carry de la primera suma
me doy cuenta si hay carry porque en caso de haberlo (asumiendo números sin signo),
el resultado sería menor a alguno de sus sumandos (que no tiene sentido al sumar números positivos)

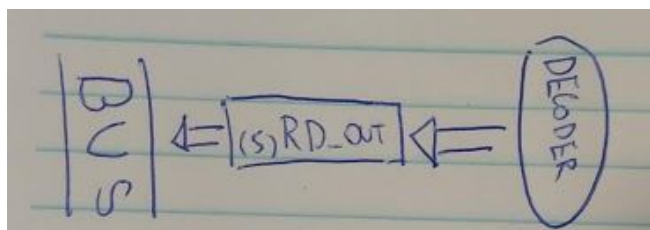
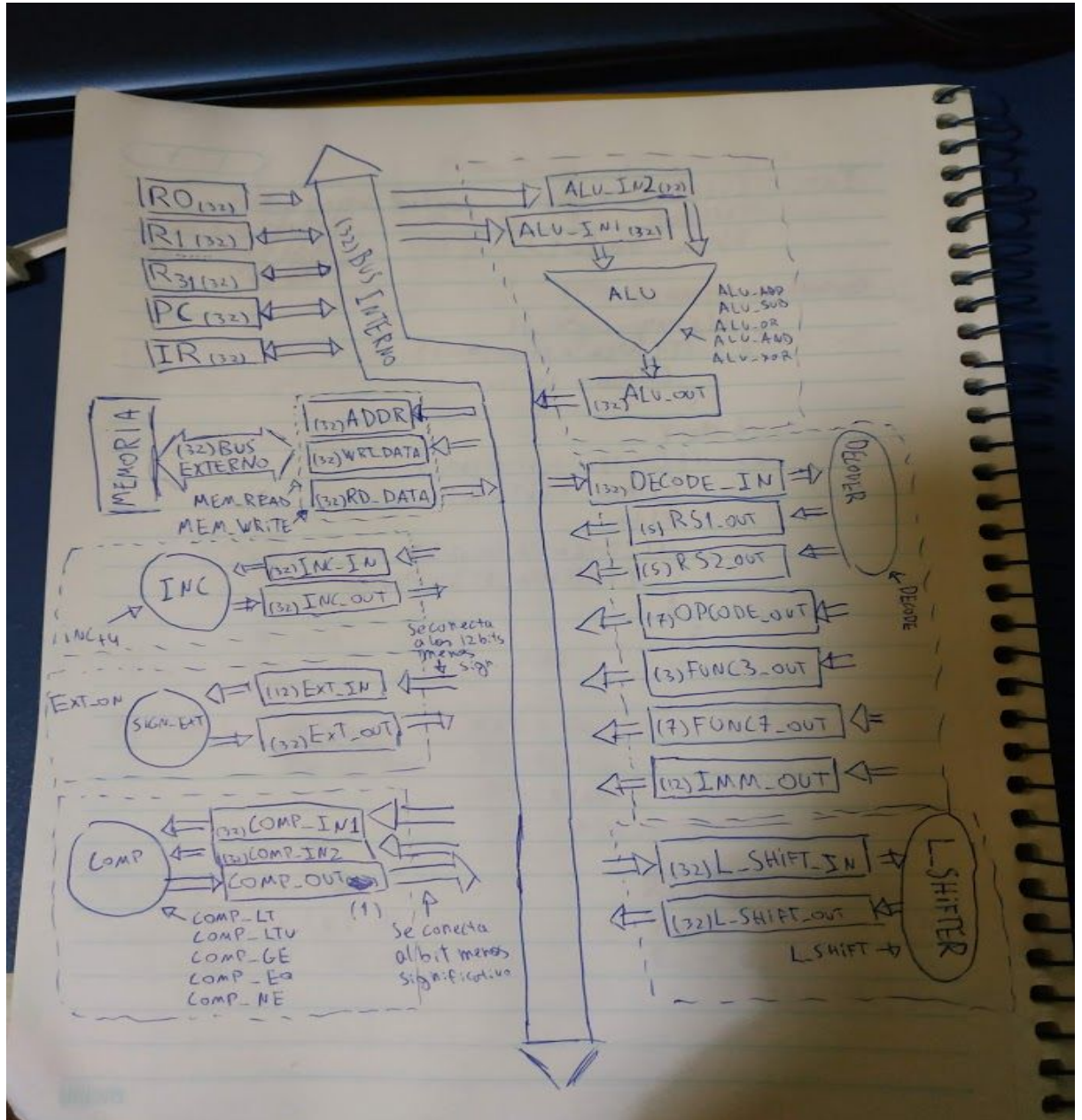
```
add r10, r12, r14  
sltu r9, r10, r12 ;me fijo si r10 es menor a r12  
add r11, r13, r15  
add r11, r11, r9 ;sumo el carry
```

2)

a)



b)



Aclaraciones:

- Todos los registros que guardan direcciones de memoria o palabras son de 32 bits
- El shifter es para salvar la excepción del bit 0 del dato Imm en las instrucciones de formato B
- Todos los registros de salida del decoder de n bits están conectados a los n bits menos significativos del bus interno.
Esto también aplica para el registro EXT_IN y COMP_OUT
- En el datapath yo dispuse todos los registros R1...31 sueltos y conectados directamente al bus por simplicidad.
- En realidad, la idea es que uno le provee un índice de registro al control unit (ese dato saldría de los registros RS1_OUT, RS2_OUT y RD_OUT del decoder).
- Luego, para leer, con un multiplexor se decide qué dato dejar salir al bus y para escribir, con un decodificador se decide qué enable_wrt habilitar, como en el taller.

c) XOR:

ALU_IN1 = RS1

ALU_IN2 = RS2

ALU_OR

RD = ALU_OUT

BGE:

COMP_IN1 = RS1

COMP_IN2 = RS2

COMP_GE

if(COMP_OUT){ //COMP_OUT es de un solo bit

ALU_IN1 = PC

EXT_IN = IMM_OUT

EXT_ON

L_SHIFT_IN = EXT_OUT //porque en imm estan los bits 1...12 y sabemos que el bit

0 es 0

L_SHIFT

ALU_IN2 = L_SHIFT_OUT

ALU_ADD

PC = ALU_OUT

ADDI:

ALU_IN1 = RS1

EXT_IN = IMM_OUT

```

EXT_ON
ALU_IN2 = EXT_OUT
ALU_ADD
RD = ALU_OUT

```

3)

a)

TEMP -> 0x00000800 (lectura)

REFR -> 0x00000804 (escritura)

1 grado -> salto de 4095/127 unidades

```

main(){
    R1 = 0x000003C7 //0°C
    R2 = 0x000008D2 //41°C
    while(true){
        if([TEMP] >= R2){
            [REFR] = 0x0000000F
        } else if([TEMP] < R1){
            [REFR] = 0x00000000
        }
    }
}

```

main:

```

R1 = 0x000003C7
R2 = 0x000008D2
R3 = 0x00000800 ;TEMP
R4 = 0x00000804 ;REFR

```

loop:

```

LW R5, 0(R3) ;uso R5 para hacer operaciones simples
BGE R5, R2, encender
BLT R5, R1, apagar
BEQ R0, R0, loop

```

encender:

```

ADDI R5, R0, 0x00F
SW R5, 0(R4)
BEQ R0, R0, loop

```

apagar:

```

ADDI R5, R0, 0x000
SW R5, 0(R4)
BEQ R0, R0, loop

```

b)

rai(){

```

    [REFR] = 0x00000000

```

}

main(){

```

    R1 = 0x000003C7 //0°C

```

```

    R2 = 0x000008D2 //41°C
    while(true){
        if([TEMP] >= R2){
            [REFR] = 0x0000000F
        }
    }
}
rai:
    ADDI R5, R0, 0x000
    SW R5, 0(R4)
main:
    R1 = 0x000003C7
    R2 = 0x000008D2
    R3 = 0x00000800 ;TEMP
    R4 = 0x00000804 ;REFR
loop:
    LW R5, 0(R3) ;uso R5 para hacer operaciones simples
    BGE R5, R2, encender
    BEQ R0, R0, loop
encender:
    ADDI R5, R0, 0x00F
    SW R5, 0(R4)
    BEQ R0, R0, loop

```