

Finales

Un programa declara un arreglo de 16 elementos de 32 bits, pasa vía stack la dirección de este arreglo a una subrutina que devolverá también vía stack el promedio (redondeado) de estos 16 valores, considera que son enteros sin signo. Este promedio es guardado en la posición C2B10233h al terminar

```

• begin
• org 2048
direccion      A . equ      C2B10h
Final          B . equ      233h
{
  sethi A, %r1
  sll %r1, 2, %r1
  add %r1, B, %r1
}
• macro push reg
  add %r14, -4, %r14
  st reg, %r14
• endmacro
• macro pop reg
  ld %r14, reg
  add %r14, 4, %r14

```


push [array] ! cargo la dir del array

push [tamaño] ! cargo el tamaño

call calcular-promedio ! llamo la sub

pop r3

st r3, r1

halt

array =

tamaño = 16

Calcular-promedio: pop r4 ! tamaño

pop r5 ! dir array

add r0, r0, r6 ! mi acumulador

call for

for:

addec r4, -1, r4

bneg fin

ld r5, r7 ! r7 tendrá mi valor actual del array

add r7, r6, r6 ! acumulo

add r5, 4, r5

ba for

fin:

srr r5, 4, r5

! como 16 es potencia de 2

push r5

$2^4 = 16$

jmp r5+4, r0

! al hacer comentario a derecha divido x 2

• end

Ej 2)

.begin

.org 2048

cte .equ 19AA9h

padron .equ 104694h

2048 - add r0, cte, r20

2052 sub r20, padron, r10

2056 st r10, [num]

2060 ld r10, [num], r5

2064 nor r10, r20, r0

2068 add r10, r20, r0

2072 jpl r15+4, r0

2076 num .dwb 1

.end

identificar que instrucción se está ejecutando el PC contiene el 80Ch

$$80Ch = 1000\ 0000\ 1100 = 2060$$

es la instrucción:

ld r10, [num], r5

la instrucción es

$\underbrace{11}_{OP} \ 0010 \ \underbrace{000000}_{OP2} \ 01010 \ 1 \ 01000 \ 0001 \ 1100$

decodifico

$$111 \ 000000 \ 00 = 1192$$

a) microcódigo

1792. $R[temp0] \leftarrow add(R[rsl], R[rs2])$

IF $R[tr[13]]$ Then goto 1794

1793. $R[rd] \leftarrow and(R[temp0], R[temp0])$

Read, goto 2047;

1794. $R[temp0] \leftarrow sext13(R[r])$

1795. $R[temp0] \leftarrow add(R[rsl], R[temp0])$

GOTO 1793

b) segunda micro

$R[rd] \leftarrow and(R[temp0], R[temp0])$

Read; GOTO 2047

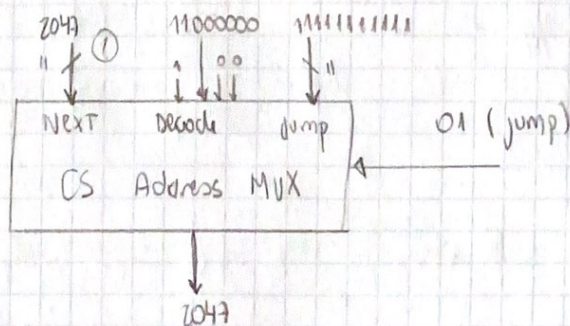
mis

A	Am	B	Bm	C	Cm	RW	AW	cond	JMPI Addr
100001	0	100001	0	000000	1	10	0101	110	1111111111

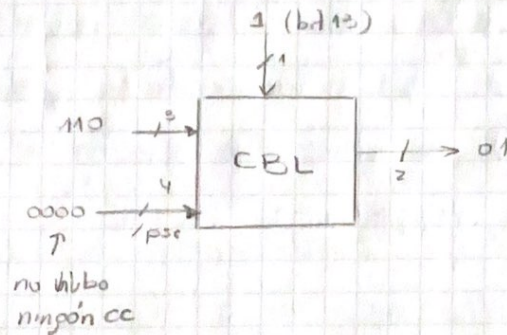
tr

11 00101 000000 01010 1 01000 0010001100

entrada y salida del multiplexor de control



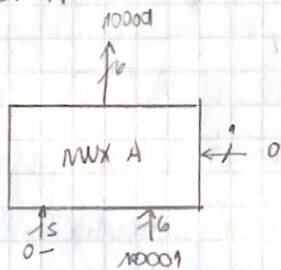
• entrada y salida de la lógica de control de saltos



Registro 1120 y 1120

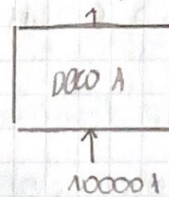
las constantes exceden los 13 bits → falla

• Multiplexor A

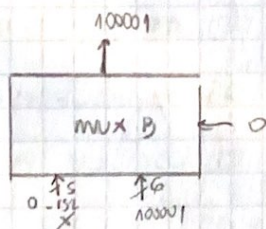


Decodificador A

todos 0 y el bit 32 en 1

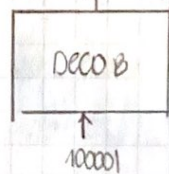


Multiplexor B

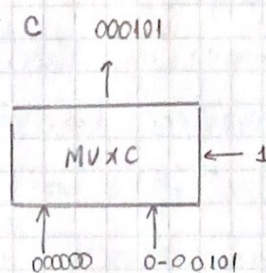


Decodificador B

todos 0 y el bit 32 en 1

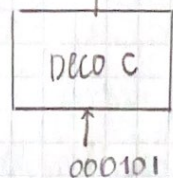


Multiplexor C



Decodificador C

todos 0 y el bit 4 en 1



- 3) Justificar la conveniencia de utilizar técnicas de caché especulada para datos y para instrucciones

Utilizar las técnicas de caché puede aumentar la velocidad de ejecución del programa dado que el acceso a los bloques de caché es más rápido que a los bloques de RAM.

- 4) Plantear una discusión respecto de la veracidad o FALSEDAZ de la siguiente afirmación

"La utilización de macros durante la escritura del código puede ~~producir~~ conducir a una aplicación cuyo tiempo de ejecución es algo más lento que el mismo código sin macros"

La utilización de macros no afecta el tiempo de ejecución, dado que si reemplazará a mano el código de la macro, el tiempo de ejecución sería el mismo. Distinta resultaría la discusión en caso de hablar de conveniencia entre macros y subrutinas. En ese caso, podemos afirmar que el uso de macros es más veloz que el uso de subrutinas ya que las últimas requieren de acceso a memoria. Sería una excepción si la subrutina se utilizara tantas veces que fuera copiada en caché; por lo tanto terminaría resultando más veloz que un acceso secuencial. En caso, contrario siempre la macro es más veloz que la subrutina. En fin, lo que quiero destacar es que las macros

no afectan el tiempo de ejecución del programa.

Por lo tanto, considero falsa la afirmación planteada.

5) Plantear una discusión respecto de la veracidad o falsedad de la siguiente "la tecnología de puentes ("bridges") no es compatible la de "bus del sistema")

La tecnología de puentes no es incompatible con la del bus del sistema.

El objetivo del bus es interconectar componentes del sistema, mientras el objetivo de la tecnología de puentes es dividir los componentes más rápidos de los más lentos. Mientras que los puentes se encargan de controlar la comunicación entre ellos, la transferencia de datos puede seguir siendo mediante buses.