



Interrogación 2

Pregunta 1

- a) ¿Por qué motivo no está soportada la instrucción ADD B, LIT en el computador básico? (1 pto.)

Solución: Al estar conectado el bus de literales y el registro B al MUX B, no es posible utilizarlos simultáneamente en una operación aritmética o lógica.

- b) ¿Aproximadamente cuántos transistores/relés utiliza el computador básico? Justifique sus cálculos y respuesta. (1 pto.)

Solución: Basándose en la Ley de Moore, es posible estimar la cantidad de transistores/relés del computador básico, si se asume que este es similar en complejidad al primer procesador de 8 bits de Intel, el 8008. En el gráfico de la Ley de Moore, el primer procesador es el 4004, aparecido en 1971 y con 2300 transistores, por lo que en 18 meses, el mejor procesador debería aumentar sus cantidad de transistores en 2300. Del mismo gráfico se puede estimar que alrededor de 1972 se lanzó el procesador 8008, por lo que el tiempo transcurrido fue de 12 meses, lo que implica un aumento de aproximadamente 1500 transistores. Luego, es posible estimar la cantidad de transistores del procesador 8008, y del computador básico en 3800.

- c) ¿Cómo se puede modificar la microarquitectura del computador básico para que este soporte direccionamiento indirecto con registro base + offset? (1 pto.)

Solución: Para soportar este modo de direccionamiento, es necesario tomar el bus de salida del registro B y conectarlo con un ADDER, que en su otra entrada recibe el valor del bus de literales. El resultado de este ADDER se agrega como otra entrada al MUX ADDRESS.

- d) Describa los tipos de datos soportados nativamente por el computador básico y por la arquitectura x86. (1 pto.)

Solución: El computador básico soporta como tipo de dato nativo números enteros y naturales binarios de 8 bits, mientras que la arquitectura x86 soporta número enteros y naturales binarios de 8 y 16 bits. Ninguna de las dos soporta números de punto flotante de forma nativa.

- e) Describa y ejemplifique los distintos tipos de **formato de instrucción** utilizados en el computador básico. (1 pto.)

Solución: El computador básico posee tres formatos de instrucción distintos:

- Opcode + 2 operandos: Por ejemplo MOV A,B
- Opcode + 1 operando: Por ejemplo PUSH B
- Opcode sin operandos: Por ejemplo RET

f) Describa en detalle la unidad de salto que se encuentra en el interior de la unidad de control del computador básico. **(1 pto.)**

Solución: La unidad de salto se encarga de cargar el registro PC con el valor del bus de literales en caso que se cumpla la condición buscada. La implementación se puede realizar con una serie de caminos lógicos alternativos, uno por cada condición, que entran a un MUX de nueve entradas, donde ocho de estas son para las condiciones y una fijada en 0 lógico. La salida de este MUX va conectada a la señal Lpc del PC, mientras que los bits de control del mux se setean de acuerdo a la instrucción de salto a ejecutar. Las condiciones se pueden agrupar y describir de la siguiente manera:

- JEQ y JNE: Para JEQ, se conecta el bit del registro Status correspondiente al flag Z al MUX. En el caso de JNE se conecta al MUX la negación de este valor.
- JGT y JLE: Para JGT, los bits correspondientes a N y Z son negados y luego conectados a un AND, cuya salida se conecta al MUX. En el caso de JLE, se conecta al MUX la negación de este valor.
- JGE y JLT: Para JGE, el bit correspondiente a N es negado y luego conectado al MUX. En el caso de JLT, se conecta al MUX la negación de este valor.
- JCR: El bit correspondiente a C es conectado al MUX.
- JOV: El bit correspondiente a V es conectado al MUX.

Pregunta 2

En las siguientes preguntas, cualquier detalle de implementación o aspecto que se asuma debe quedar claramente explicado.

- a) Escriba en el assembly del computador básico un programa que calcule el n-ésimo número de Fibonacci. **(3 ptos.)**

Solución:

```
DATA:
n ?
index 0
prev1 0
prev2 1
res 0

CODE:
MOV A,(n)
CMP A,0
JEQ end0

while:
MOV A,(n)
MOV B,(index)
CMP A, B
JEQ end
INC B
MOV (index), B
MOV A, (prev1)
MOV B, (prev2)
ADD A, B
MOV (prev1), B
MOV (prev2), A
MOV (res), A
JMP while

end0:
MOV A, 1
MOV (res), A
end:
```

- b) Escriba en assembly x86, usando al menos una subrutina y la convención *stdcall*, un programa que calcule la raíz cuadrada natural de un número natural utilizando el método de Newton. Este método aproxima iterativamente el valor de x, tal que $f(x) = 0$, mediante la siguiente recurrencia: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$. **(3 ptos.)**
Hint: defina $f(x) = x^2 - n$.

Solución:

```
MOV AX, 0
MOV AL, n
PUSH AX
CALL newton
MOV res, AL
RET

newton:
PUSH BP
MOV BP, SP

MOV BX, [BP+4] ;BX = n
MOV CX, BX      ;CX = x_i = n

do_while:
MOV AX, BX
DIV CL          ;AL = n/x_i
ADD AX, CX      ;AL += x_i + n/x_i
SHR AL, 1       ;AL = (x_i+n/xi)/2
CMP AL, CL      ;revisamos si hay cambio
JE end
MOV CL, AL      ;x_{i+1} = x_i
JMP do_while

end:
MOV AX, CX
RET 2

n      db      ?
res    db      0
```

Pregunta 3

Se desea modificar el computador básico a nivel de microarquitectura e ISA. Para los siguientes puntos detalle las modificaciones que haría. Utilice diagramas de componentes y conexiones y tablas de opcodes e instrucciones cuando corresponda.

- a) Tener stack de uso general e independiente de la memoria de datos. **(3 ptos.)**
- b) Dar soporte para la instrucción SUBLEQ A, B, Dir, definida de la siguiente manera:
 - $\text{Mem}[A] = \text{Mem}[A] - \text{Mem}[B]$
 - if $(\text{Mem}[a] \leq 0) \Rightarrow \text{JMP Dir}$

Su implementación no debe sobrescribir datos en cálculos intermedios. **(2 ptos.)**

- c) Aumentar el tamaño del bus de direcciones a 16 bits, manteniendo el resto de los tamaños del computador en 8 bits. Debe ponerse especial énfasis en las instrucciones para direccionar la memoria. **(2 ptos.)**