

Guía de Trabajos Prácticos N° 3

Microarquitectura del procesador

Teoría

1. Considere la siguiente instrucción

Instrucción: `and rd, rs1, rs2`

Interpretación: `Reg[rd] = Reg[rs1] AND Reg[rs2]`

- ¿Cuáles son los valores de las señales de control generadas por el control en el diagrama de bloques de la microarquitectura monociclo para esta instrucción?
- ¿Qué recursos (bloques) realizan una función útil para esta instrucción?
- ¿Qué recursos (bloques) no producen salida para esta instrucción? ¿Qué recursos producen resultados que no se utilizan?

2. Considere un conjunto de instrucciones compuesto por los siguientes tipos de instrucciones

Tipo R	Tipo I	Load	Store	Branch	Jump
24%	28%	25%	10%	11%	2%

- ¿Qué fracción de todas las instrucciones usan memoria de datos?
 - ¿Qué fracción de todas las instrucciones usa la memoria de instrucciones?
 - ¿Qué fracción de todas las instrucciones usan la unidad de generación de datos?
 - ¿Qué fracción de todas las instrucciones usan la unidad de generación de datos inmediatos?
 - ¿Qué hace la unidad de generación de datos durante los ciclos en los que no se necesita su salida?
3. Cuando se fabrican chips de silicio, los defectos en los materiales y los errores de fabricación pueden resultar en circuitos defectuosos. Un defecto muy común es que un cable de una señal se rompa y siempre se registre un 0. Esta falla se denomina "**stuck-at-0**".
- ¿Qué instrucciones no funcionan correctamente si el bit más significativo de **Data_S** está atascado en 0?
 - ¿Qué instrucciones no funcionan correctamente si el bit más significativo de **Jump** está atascado en 0?
 - ¿Qué instrucciones no funcionan correctamente si **ALU_S** = 0?
4. A continuación examinamos en detalle cómo se ejecuta una instrucción en el Datapath de una microarquitectura monociclo. El procesador obtiene la siguiente palabra de instrucción

0x00c6ba23.

de la memoria de programa.

- ¿Cuáles son los valores de **ALU_Op** para esta instrucción?

- b. ¿Cuál es la nueva dirección de PC después de que se ejecuta esta instrucción? Resalte la ruta a través de la cual se determina este valor.
- c. Para cada multiplexor muestre los valores de sus entradas y salidas durante la ejecución de esta instrucción. Enumere los valores que son salidas de registro.
- d. ¿Cuáles son los valores de entrada para la ALU y las dos unidades de suma?
- e. ¿Cuáles son los valores de todas las entradas para la unidad de registros?

5. Evalúe la dificultad de agregar la instrucción **Load With Increment** a nuestro procesador.

Instrucción: `lwi.d rd, rs1, rs2`

Interpretación: `Reg[rd]=Mem[Reg[rs1]+Reg[rs2]]`

- a. ¿Qué nuevos bloques funcionales (si hiciera falta) necesitamos para esta instrucción?
- b. ¿Cuales de los bloques funcionales existentes (si hiciera falta) requieren modificación?
- c. ¿Qué nuevas rutas de datos (si hiciera falta) necesitamos para esta instrucción?
- d. ¿Qué nuevas señales necesitamos (si hiciera falta) de la unidad de control para respaldar esta instrucción?

6. Evalúe la dificultad de agregar la instrucción **Swap** a nuestro procesador.

Instrucción: `swap rs1, rs2`

Interpretación: `Reg[rs2]=Reg[rs1]; Reg[rs1]=Reg[rs2]`

- a. ¿Qué nuevos bloques funcionales (si hiciera falta) necesitamos para esta instrucción?
- b. ¿Cuales de los bloques funcionales existentes (si hiciera falta) requieren modificación?
- c. ¿Qué nuevas rutas de datos (si hiciera falta) necesitamos para esta instrucción?
- d. ¿Qué nuevas señales necesitamos (si hiciera falta) de la unidad de control para respaldar esta instrucción?
- e. Modifique el diagrama de bloques de la implementación monociclo para demostrar una implementación de esta nueva instrucción.

7. Supongamos que los bloques lógicos utilizados para implementar el procesador tienen las siguientes latencias:

Banco de Registros	Registro	Multiplexor	ALU	Sumador	Gen. Dato Inmediato	Gen. Dato	Op. en Memoria	Control
50 ps	150 ps	25 ps	200 ps	150 ps	50 ps	70 ps	250 ps	50 ps

- a. ¿Cuál es la latencia de una instrucción de tipo R, es decir, cuánto tiempo debe ser el período de reloj para garantizar que esta instrucción funcione correctamente?
- b. ¿Cuál es la latencia de **ld**?
- c. ¿Cuál es la latencia de **sd**?
- d. ¿Cuál es la latencia de **beq**?
- e. ¿Cuál es la latencia de una instrucción de tipo I?

f. ¿Cuál es la frecuencia máxima de reloj mínimo para este procesador?

8. En esta pregunta examinamos cómo la segmentación (pipelining) afecta el ciclo de reloj del procesador. Supongamos que las etapas individuales de la ruta de datos tienen las siguientes latencias

Instr. Fetch (IF)	Dec. de Inst (ID)	Ejecución (EX)	Oper. Fetch (MEM)	Esc. de Res. (WB)
250 ps	350 ps	150 ps	300 ps	200 ps

Además, las instrucciones ejecutadas por el procesador se desglosan de la siguiente manera:

ALU	Jump/Branch	Load	Store
45%	20%	20%	15%

- ¿Cuál es el tiempo de ciclo del reloj en un procesador con y sin ejecución segmentada?
 - ¿Cuál es la latencia total de una instrucción **ld** en un procesador con y sin ejecución segmentada?
 - Si podemos dividir una etapa de la ruta de datos segmentada en dos nuevas etapas, cada una con la mitad de la latencia de la etapa original, ¿qué etapa dividiría y cuál es el nuevo tiempo del ciclo de reloj?
 - Suponiendo que no hay detenciones (stalls) ni riesgos (hazards), ¿cuál es la utilización de la memoria de datos?
 - Suponiendo que no hay detenciones (stalls) ni riesgos (hazards), ¿cuál es la utilización del puerto de escritura de la unidad "Registros"?
9. Suponga que x11 se inicializa en 11 y x12 se inicializa en 22. Suponga que ejecutó el siguiente código en un procesador segmentado que no maneja riesgos de datos (es decir, el programador es responsable de abordar los riesgos de datos insertando instrucciones NOP donde sea necesario). ¿Cuáles serían los valores finales de los registros x13 y x14?

```
addi x11, x12, 5
add x13, x11, x12
addi x14, x11, 15
```

10. Suponga condiciones de ejecución similares a la de la pregunta 9. ¿Cuáles serían los valores finales del registro x15? Suponga que el archivo de registro se escribe al comienzo del ciclo y se lee al final de un ciclo. Por lo tanto, una etapa de ID devolverá los resultados de un estado WB que ocurra durante el mismo ciclo.

```
addi x11, x12, 5
add x13, x11, x12
addi x14, x11, 15
add x15, x11, x11
```

11. Si cambiamos las instrucciones de carga/almacenamiento para solo usar un registro (sin un desplazamiento) como dirección, estas instrucciones ya no necesitan usar la ALU. Como resultado, las etapas MEM y EX pueden superponerse y la tubería tiene solo cuatro etapas.
- ¿Cómo afectará la reducción de la profundidad de la segmentación al tiempo del ciclo?
 - ¿Cómo podría este cambio mejorar el rendimiento de la segmentación?
 - ¿Cómo podría este cambio degradar el rendimiento de la segmentación?

12. Consideremos el siguiente programa

```
LOOP:    ld x10, 0(x13)
         ld x11, 8(x13)
         add x12, x10, x11
         subi x13, x13, x16
         bnez x12, LOOP
```

Suponiendo que se utiliza un predictor de salto perfecto (sin paradas debido a riesgos de control), que no hay detenciones, que la segmentación tiene soporte de adelanto completo y que los saltos incondicionales se resuelven en la etapa EX, en lugar de ID.

- Desarrolle un diagrama de ejecución segmentada para las dos primeras iteraciones del bucle.
- Marque las etapas del pipeline que no realizan un trabajo útil. ¿Con qué frecuencia, mientras el pipeline está lleno, tenemos un ciclo en el que las cinco etapas del pipeline están haciendo un trabajo útil? Comience el ciclo cuando la instrucción **subi** está en la etapa IF y termine con el ciclo cuando la instrucción **bnez** está en la etapa IF.

13. Las preguntas de este ejercicio se refieren a la siguiente secuencia de instrucciones y suponen que se ejecuta en un datapath segmentado en cinco etapas:

```
add x15, x12, x11
ld x13, 4(x15)
ld x12, 0(x2)
or x13, x15, x13
sd x13, 0(x15)
```

- Si no hay unidad de adelanto o detección de riesgo, inserte NOP para garantizar una ejecución correcta.
- Reorganice el código para minimizar la cantidad de NOP necesarios. Puede asumir que el registro x17 se puede usar para contener valores temporales en su código modificado.
- Si el procesador tiene unidad de adelanto, pero nos olvidamos de implementar la unidad de detección de riesgos ¿Qué sucede cuando se ejecuta el código original?
- Si hay unidad de adelanto, especifique qué señales se generan en cada ciclo por las unidades de adelanto y detección de riesgos durante la ejecución de los primeros siete ciclos de este código.
- Si no hay unidad de adelanto ¿Qué nuevas señales de entrada y salida necesitamos para la unidad de detección de riesgos? Utilizando esta secuencia de instrucciones como ejemplo, explique por qué se necesita cada señal.

14. La importancia de tener un buen predictor de saltos depende de la frecuencia con la que se ejecutan los saltos condicionales. Junto con la precisión del predictor, esto determinará cuánto tiempo se pasa estancando debido a saltos mal predichos. Suponga que el desglose de las instrucciones dinámicas en varias categorías de instrucciones es el siguiente:

Tipo R	beqz/bnez	jal	ld	sd
40%	25%	5%	25%	5%

La precisión de los predictores es la siguiente

Se toma siempre	Nunca se toma	2 -bits
45%	55%	85%

- Los ciclos de detención (burbujas y flush) debido a saltos mal predichos aumentan la cantidad de ciclos de reloj por instrucción (CPI). ¿Cuál es el CPI adicional debido a los saltos mal predichos con el **predictor se-toma-siempre**? Suponga que los resultados de la rama se determinan en la etapa de decodificación (ID) y se aplican en la etapa de ejecución (EX) que no hay riesgos de datos.
- Repita inciso a para el predictor **Nunca-se-toma**.
- Repita inciso a para el predictor de 2 bits.
- Con el predictor de 2 bits, ¿qué aceleración se lograría si pudiéramos convertir la mitad de las instrucciones de bifurcación en alguna instrucción ALU? Suponga que las instrucciones predichas correcta e incorrectamente tienen la misma probabilidad de ser reemplazadas.
- Con el predictor de 2 bits, ¿qué aceleración se lograría si pudiéramos convertir la mitad de las instrucciones de bifurcación de una manera que reemplazara cada instrucción de bifurcación con dos instrucciones ALU? Suponga que las instrucciones predichas correcta e incorrectamente tienen la misma probabilidad de ser reemplazadas.
- Algunas instrucciones de bifurcación son mucho más predecibles que otras. Si sabemos que el 80% de todas las instrucciones de salto cuya dirección final está antes de la dirección actual siempre se predicen correctamente, ¿Cuál es la precisión del predictor de 2 bits en el 20% restante de las instrucciones de salto?