

## Ejercicio ruta de datos multiciclo

Nos piden incluir tres nuevas instrucciones en el procesador MIPS multiciclo de la última clase de teoría (ALU y memoria unificadas). Para cada caso debemos indicar:

- Modificaciones en la ruta de datos si son necesarias (por ejemplo incluyendo un nuevo multiplexor o un doble puerto de escritura en BR)
- Acciones RTL (transferencia entre registros)
- Valor de las señales de control en cada ciclo. Basta con especificar aquellas cuyo valor sea relevante.
- Camino crítico y tiempo de ciclo
- Tiempo de ejecución (Tex) suponiendo que el programa de referencia se compone de un 10% sw, 20% lw, 50% aluops, 8% beq tomados, 2% no tomados, 5% JAL, 5% RET
  - $\text{Tex} = \text{Ciclos totales} \times \text{Tc}$
  - **Hint:** Ciclos totales =  $\sum \text{ciclos\_instruccion\_tipo}_i \times \text{coef\_unitario\_instr\_tipo}_i$ 
    - **Ej:** Si sw tarda 4 ciclos, contribuye con  $4 \times 0.1$
- **Delays:** registros: tsup = td = 1 ps. RF: 10 ps. ALU+4: 2ps. Mux2-1= Mux2-1ctrl = Mux4-1ctrl = 2 ps. Mux4-1 = 3 ps. Memoria: tsup 2 ps; td = 20 ps. ALU = 10 ps. UC Moore: FT: 4ps. FS: 4 ps. Status register: ts = td = 1 ps

### Caso 1: lw, (rs, LSR in<sub>m0-1</sub>) Load indirecto con post-incremento

Este lw accede a la dirección indicada por Rs, y después actualiza Rs para que señale a la siguiente dirección:

```
Lw rt, [rs]#inm :
BR(rt) ← Mem(BR(rs));
BR(rs) ← BR(rs) + SignExt(inm16);
PC ← PC + 4;
```

### Caso 2: jal rt, imm16 (format I)

```
BR(rt) ← PC + 4;
PC ← PC + 4 + 4*SignExt(inmed16);
```

### Caso 3: ret rs (format I)

```
PC ← BR(rs);
```

## Metodología

1. Repasar /estudiar la ruta completa adjunta, vista más o menos completa en clase.  
Adjuntamos abajo las acciones RTL
  - Anotar los valores de las señales de control necesarias para cada etapa de cada instrucción
2. Detallar qué acciones RTL tengo que llevar a cabo en cada etapa para las nuevas instrucciones que se piden en este problema
3. Ir a la ruta de datos y ver qué buses, muxes y señales de control debo añadir para poder realizar las nuevas acciones.
4. Ajustar las acciones a las etapas de la instrucción e indicar las señales de control necesarias.

## RTL ruta multiciclo

### add rd, rs, rt

#### Fetch

IR<- M(PC)

PC <- PC+4

#### Decode

A<-BR(rs)

B<-BR(rt)

#### Execution

ALUout <- A + B

#### Memory

---

#### Write

BR(rd) <- ALUOut

### lw rt, imm(rs)

#### Fetch

IR<- M(PC)

PC <- PC+4

#### Decode

A<-BR(rs)

#### Execution

ALUout <- A + sex(imm)

#### Memory

MDR<-M(ALUout)

#### Write

BR(rt) <- MDR

### sw rt, imm(rs)

#### Fetch

IR<- M(PC)

PC <- PC+4

#### Decode

B<-BR(rt)

#### Execution

ALUout <- A + sex(imm)

#### Memory

M(ALUout) <- B

#### Write

### beq rs rt imm

#### Fetch

IR<- M(PC)

PC <- PC+4

#### Decode

A<-BR(rs)

B<-BR(rt)

#### Execution

Z <- A -B

#### Memory

---

#### Write

PC <- PC + sex(imm)\*4

