

# ARQUITECTURA DE COMPUTADORES

## 543.426

AYUDANTE: ANTONIO SAAVEDRA

Ayudantía de pipeline  
27 de mayo de 2017

### 1. Problema 1

Suponga un procesador MIPS segmentado en un pipeline de 8 etapas: 2 de fetch, 1 de decodificación, 2 de ejecución, 2 de acceso a memoria de datos y una de escritura en registros. El pipeline está regularizado, utiliza saltos retrasados que se resuelven en la etapa de decodificación, y soporta forwarding de datos en todas las etapas. Considere el siguiente código:

```
1 I1: lw    $t1, 0($a0)
2 I2: lw    $t2, 1000($a0)
3 I3: add   $t1, $t1, $t2
4 I4: sw    $t1, 2000($a0)
5 I5: addi  $a0, $a0, 4
6 I6: bne   $a0, $a1, I1
7 I7: nop
8 I8: nop
```

- Identifique las dependencias RAW, WAR, y WAW dentro de una ejecución de este código.
- Grafique la ejecución del código y calcule su CPI sostenido.
- Calcule el CPI sostenido graficando luego de reordenar el código para minimizar burbujas. Renombre registros y ajuste constantes de ser necesario.
- Calcule el CPI sostenido graficando luego de aplicar loop unrolling x2, renombrar registros y reordenar el código para optimizar su desempeño. Suponga que el lado itera un número par de veces.

## 2. Problema 2

En un procesador MIPS con un pipeline en 12 etapas se pretende ejecutar el siguiente código:

```
1  I1: lw    $t1, 0($a0)
2  I2: add   $t1, $t1, $t0
3  I3: lw    $t2, 40($a0)
4  I4: add   $t2, $t2, $t0
5  I5: mul   $t2, $t1, $t2
6  I6: lw    $t1, 0($a1)
7  I7: sub   $t1, $t2, $t1
8  I8: add   $t2, $t1, $t2
9  I9: sw    $t1, 4($a1)
10 I10: addi  $a1, $a1, 4
11 I11: add   $a0, $a0, $t2
12 I12: beq   $a1, $t5, I1
```

Las etapas del pipeline se descomponen en 2 etapas de fetch, 2 etapas de decodificación, 3 de ejecución en ALU, 4 de acceso a memoria de datos y 1 de escrita en los registros. El pipeline está regularizado y ejecuta los saltos utilizando un predictor de saltos que se asume perfecto y los resuelve al finalizar el segundo ciclo de la etapa de decodificación. El pipeline soporta forwarding en todas sus etapas.

a) Identifique las dependencias RAW, WAR, y WAW dentro de una ejecución de este código.

b) Calcule el CPI sostenido graficando luego de reordenar el código para minimizar burbujas. Renombre registros y ajuste constantes de ser necesario.