Arquitectura del Procesador II

Práctico U1.1 - Programación del MIPS

Ejercicio 1: Analice el conjunto de instrucciones del procesador MIPS. Luego, realice la traducción del siguiente código de alto nivel al lenguaje de ensamble de este procesador:

```
C = A + B;
E = D[7] – A;
C = E + A * 32;
```

Ejercicio 2: Traduzca el siguiente código en alto nivel al lenguaje de ensamble del MIPS:

```
if(C==0){
    A = B + D[12];
}else{
    A = B++;
}
```

Ejercicio 3: Traduzca el siguiente código en alto nivel al lenguaje de ensamble del MIPS:

```
for(I=0; I<100; I++){
    A[I] += A[I];
}
```

Ejercicio 4: Traduzca el siguiente código de alto nivel al lenguaje de ensamble del MIPS:

```
for(I=0; I<4; I++){
    A[I] = B[I] * 8;
}
```

Ejercicio 5: Para cada una de las instrucciones indique el tipo de instrucción que le corresponde y realice la codificación en lenguaje máquina:

add $t0, $s1, $s2	lw $t0, 16($s3)	sw $t1, 17($s2)	sll $t2, $s0, 4	bne $s3, $s4, 33
j 0x1000	andi $s3, $s3, 1	slt $t0, $s3, $s4	jr $ra	jal 0x1000

Ejercicio 6: Para cada una de las instrucciones del ejercicio anterior indique sobre el camino de datos del MIPS cuales son los caminos y las unidades funcionales utilizadas durante su ejecución.

Ejercicio 7: Realice la traducción del siguiente código en alto nivel al lenguaje de ensamble del MIPS:

```
for(I=0; I<1000; I++){
    A[I] = B[I] + 67;
}
```

Ejercicio 8: Si usted necesita trabajar con una constante de más de 16 bits, por ejemplo $(F1F2F3F4)_{16}$, claramente no es posible colocarla dentro del valor inmediato de una instrucción aritmética o lógica, ¿Cómo resolvería esta dificultad?.

Ejercicio 9: ¿Cómo se realiza el llamado y retorno a subrutinas en el MIPS?.

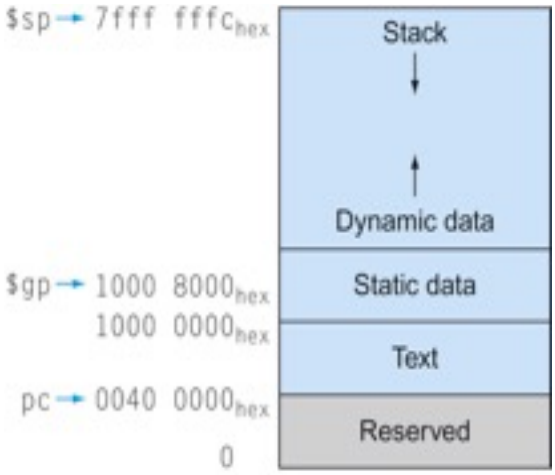
Ejercicio 10: ¿Qué es una pseudoinstrucción?.

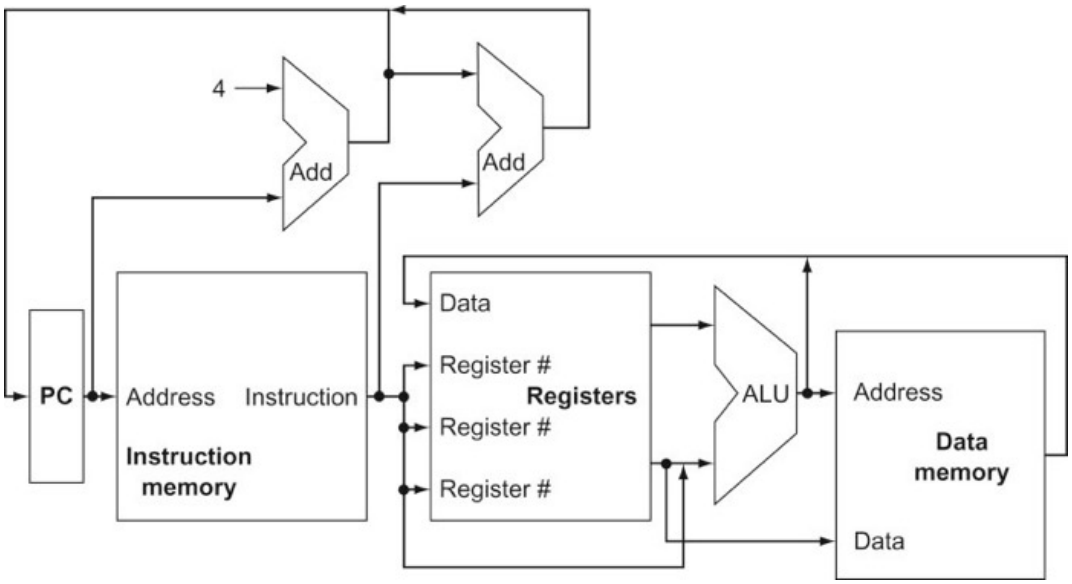
Ejercicio 11: ¿Cómo se traducen las siguientes pseudoinstrucciones?.

mov $a0, $a1	(Move)
li $a1, 0xFA00	(Load Immediate)
li $a0, 0xF000A001	(Load Immediate)
la $t0, 0xFF00	(Load Address)
la $t0, 0xFFFF0000	(Load Address)
lw $t1, 0xFFFF01($a0)	(Load Word)

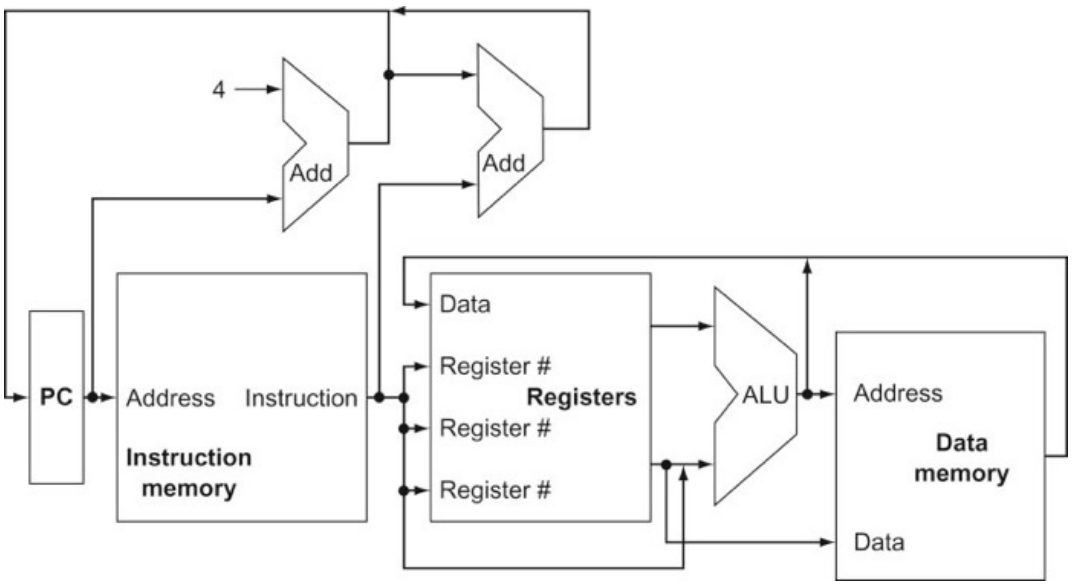
Ejercicio 12: Analizando el Datapath del Mips, responda:
1. ¿Cuántos bits posee el registro PC?
2. ¿Qué circuito es necesario para realizar la actualización del registro PC?
3. ¿Por qué es necesario tener 2 memorias (una de datos y otra de instrucciones)?
4. ¿Cómo se construye un banco de registros con dos puertos de lectura?
5. ¿Por qué se realiza un shift a izquierda de dos lugares al offset de las instrucciones de branch antes de ser sumado al contenido registro PC?

Ejercicio 13: En el MIPS una de las convenciones odoptadas es inicializar el registro usado como puntero global $gp (registro 28) con el valor 0x10008000, para permitir el acceso a datos estáticos y dinámicos. Las instrucciones de acceso a memoria que usan como base $gp pueden acceder a una porción de la memoria. Se pide especificar cuatro instrucciones load word que recuperen la palabra más alejada y la más cercana por encima y por debajo de la base $gp.





MIPS - Camino de datos simple



MIPS - Camino de datos simple