

	<p align="center"> UNIVERSIDAD DE COSTA RICA ESCUELA DE INGENIERÍA ELÉCTRICA Ejercicios de Práctica </p>	<p align="center"> EIE <hr/> Escuela de Ingeniería Eléctrica </p>
	<p align="center"> IE-0321 Estructuras de Computadores Digitales I Profesor: Jose Ariel Fallas Pizarro </p>	

- 1) Escriba en lenguaje ensamblador de Mips el siguiente código que está en alto nivel.

```
for (i=1; i<129;i=i*2)
{
    A[i]=A[i-1]+A[i+1];
}
```

suponga que i se mapea en \$t0 y A en \$s0.

- 2) Escriba en lenguaje ensamblador de Mips el siguiente código que está en alto nivel.

```
i=0;
while (A[2*i] != 0)
{
    A[i]=A[2*i-1]+A[2*i+1];
    i++;
}
```

suponga que i se mapea en \$t0 y A en \$s0.

- 3) Escriba en lenguaje ensamblador de Mips el siguiente código que está en alto nivel.

```
i=0;
while (A[i]>=0)
{
    for (j=0; j<i; j++)
    {
        A[i]=A[i]+A[j];
    }
    i++;
}
```

suponga que i se mapea en \$t0 y A en \$s0.

- 4) Para el procesador MIPS segmentado de 5 etapas visto en clase. Si no hay anticipación o detección de riesgos, en el siguiente código inserte instrucciones nop para garantizar una ejecución correcta.

```
add $t5,$t2,$t1
lw  $t3,4($t5)
lw  $t2,0($t2)
or  $t3,$t5,$t3
sw  $t3,0($t5)
```

- 5) Para el procesador MIPS segmentado de 5 etapas visto en clase. Si no hay anticipación o detección de riesgos, en el siguiente código inserte instrucciones nop para garantizar una ejecución correcta. (Asuma el peor caso de ejecución)

```
add $t5,$t2,$t1
lw  $t3,4($t5)
beq $t3,0,jump
lw  $t2,0($t2)
jump :
add $t3,$t2,$t3
sw  $t3,0($t5)
```

- 6) Escriba, en lenguaje ensamblador de mips, una función que reciba por medio de \$a0 la dirección de un array A y por \$a1 N (el número de palabras de este array). Este array corresponde a números enteros sin signo. La función debe devolver por \$v0 el número mayor del array y por \$v1 el número menor del array.
- 7) Escriba, en lenguaje ensamblador de mips, una función que reciba por medio de \$a0 la dirección de un array A y por \$a1 N (el número de palabras de este array). Este array corresponde a números enteros sin signo. La función debe devolver por \$v0 un uno si hay dos números iguales consecutivos en el array y un cero en caso contrario.

- Calcule la cantidad de bits que se utilizan para offset
- Calcule la cantidad de bits que se utilizan como index
- Calcule la cantidad de bits que se utilizan como TAG
- Calcule la tasa de aciertos

[illegible]

- Calcule la cantidad de bits que se utilizan para offset
- Calcule la cantidad de bits que se utilizan como index
- Calcule la cantidad de bits que se utilizan como TAG
- Calcule la tasa de aciertos

[illegible]