

Solucions Problemes Encarregats (4.2, 4.9, 1.7, 1.11)

Problema 4.2 de la col·lecció

Tradueix a llenguatge ensamblador MIPS la següent funció, on N és una constant:

```
void func(int mat[][N], int i, int j) {
    if (i>j)
        mat[i][j] = mat[j][i];
}
```

```
func:    ble    $a1, $a2, fi_if
        #@mat[j][i] = @mat + (j*N + i)*4
        li     $t0, N
        mult   $a2, $t0
        mflo   $t1
        addu   $t1, $t1, $a1
        sll    $t1, $t1, 2
        addu   $t1, $t1, $a0
        lw     $t1, 0($t1)
        #@mat[i][j] = @mat + (i*N + j)*4
        mult   $a1, $t0
        mflo   $t3
        addu   $t3, $t3, $a2
        sll    $t3, $t3, 2
        addu   $t3, $t3, $a0
        sw     $t1, 0($t3)
fi_if:   jr     $ra
```

Problema 4.9 de la col·lecció

Donades les següents declaracions:

```
#define N 100
char vecchar[N];
int matint[N][N];
long long matlong[N][N];

void func() {
    int i;
    ... /* aquí va la sentència de cada apartat */
}
```

Tradueix a MIPS les següents sentències utilitzant accés seqüencial, suposant que pertanyen a la funció func, i que la variable i ocupa el registre \$t0:

a) for (i=1; i<N; i+=2) /* Atenció, la i va de 2 en 2 */
 vecchar[i] = N-i;

Solució:

```
        li     $t0, 1           #i
        la     $t1, vecchar + 1 #Punter inicial
        li     $t2, N
for:     bge    $t0, $t2, fi_for
        subu   $t3, $t2, $t0
        sb     $t3, 0($t1)
        addiu  $t1, $t1, 2       #stride
        addiu  $t0, $t0, 2
```

```

        b    for
fi_for:

```

```

b) for (i=0; i<N; i++)
        matint[i][0] = 0;

```

Solució:

```

        li    $t0, 0    #i
        la    $t1, matint                #Punter inicial
        li    $t2, N
for:     bge   $t0, $t2, fi_for
        sw    $zero, 0($t1)
        addiu $t1, $t1, N*4              #stride
        addiu $s0, $t0, 1
        b     for
fi_for:

```

```

c) i=4;
while (matint[3][i] != 0) {
    matint[3][i]--;
    i+=3;
}

```

Solució:

```

        li    $t0, 4                    #i
        la    $t1, matint + 3*N*4 + 4*4
while:   lw    $t2, 0($t1)
        beq   $t2, $zero, fi_while
        addiu $t2, $t2, -1
        sw    $t2, 0($t1)
        addiu $t1, $t1, 12              #stride
        b     while
fi_while:

```

```

d) for (i=N-1; i>=0; i--)
        matlong[i][4] = i;    /* Atenció a la part alta! */

```

Solució:

```

        li    $t0, N-1    #i
        la    $t1, matlong + (N-1)*N*8 + 4*8
for:     blt   $t0, $zero, fi_for
        sw    $t0, 0($t1)
        sw    $zero, 4($t1)
        addiu $t1, $t1, -N*8          #stride
        addiu $t0, $t0, -1
        b     for
fi_for:

```

Problema 1.7 de la col·lecció

Tenim dos implementacions diferents del mateix ISA, on hi han 4 classes d'instruccions A, B, C, D. La taula següent mostra la freqüència de relotge i el CPI de cada classe, per les dues implementacions.

| | Clock Rate | CPI A | CPI B | CPI C | CPI D |
|----|------------|-------|-------|-------|-------|
| P1 | 1.5 Ghz | 1 | 2 | 3 | 4 |
| P2 | 2 Ghz | 2 | 2 | 2 | 2 |

1. Quin seria el CPI en mitjana de cada implementació, suposant un programa de 10^6 instruccions distribuïdes de la forma : 10% Classe A, 20% Classe B, 50% Classe C i 20% Classe D?
2. Quina de les dues implementacions és més ràpida, suposant un programa 10^6 ?

SOLUCIÓ:

1. $CPI_{P1} = 0,1 * 1 + 0,2 * 2 + 0,5 * 3 + 0,2 * 4 = 2,8cpi$.
 $CPI_{P2} = 0,1 * 2 + 0,2 * 2 + 0,5 * 2 + 0,2 * 2 = 2cpi$.
2. $T_{exeP1} = 10^6 * (0,1 * 1 + 0,2 * 2 + 0,5 * 3 + 0,2 * 4) * (1/1,5) * 10^{-9} = 10^6 * 1,87 * 10^{-9} = 1,87 * 10^{-3} = 1,87ms$.
 $T_{exeP2} = 10^6 * (0,1 * 2 + 0,2 * 2 + 0,5 * 2 + 0,2 * 2) * (1/2) * 10^{-9} = 10^6 * 1,0 * 10^{-9} = 1,0 * 10^{-3} = 1ms$.

P2 és més ràpid.

Problema 1.11 de la collecció

Tot i que la potència dinàmica és la principal font de dissipació de la potència en una CMOS, la pèrdua produeix una dissipació de la potencia estàtica, de la forma $V * I_{leak}$. Quant més petit és el circuit més significativa és la potència estatica. La taula següent ens mostra la dissipació de potència estatica i dinamica per a dues generacions de processadors.

| | Tecnologia (nm) | Potència dinàmica (W) | Potència estàtica (W) | Voltatge (V) |
|---|-----------------|-----------------------|-----------------------|--------------|
| a | 250 | 49 | 1 | 3.3 |
| b | 90 | 75 | 45 | 1.1 |

1. Troba quin percentatge de potència total dissipada correspon a la potència estàtica i dinàmica en cada generació.
2. Si la potència estàtica depèn de la corrent de pèrdua (I_{leak}), $P = V * I_{leak}$. Troba la corrent de pèrdua en cada tecnologia.
3. En quin percentatge podríem reduir la potència total de cada processador reduint només la potència dinàmica?

SOLUCIÓ:

1. (a) $\%P_s = 1/50 = 2\%$; $\%P_d = 49/50 = 98\%$
(b) $\%P_s = 45/120 = 37,5\%$; $\%P_d = 75/120 = 62,5\%$
2. (a) $I_{leak} = P_s/V = 1/3,3 = 0,3A$
(b) $I_{leak} = P_s/V = 45/1,1 = 40,90A$
3. $P_1 = \lim_{P_d \rightarrow 0} P = \text{Potència estàtica}$.
(a) $\%millora = \frac{P_0 - P_1}{P_0} = \frac{50 - 1}{50} = \frac{49}{50} = 0,98 \rightarrow 98\%$ ($50/1 = 50$ vegades)
(b) $\%millora = \frac{P_0 - P_1}{P_0} = \frac{120 - 45}{120} = \frac{75}{120} = 0,625 \rightarrow 62,5\%$ ($120/45 = 2,67$ vegades)