

Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 1. (3 puntos)

Dado las siguientes rutinas escritas en C:

```
int MaxMin(int a, int *b){      int Matxaca(char v[]){      int Exa(int a, int *b){
    if (a > *b)                  int i;                      int max;
      return a;                  i=0;
    else {                      while (v[i] != '\.'){          max = MaxMin(a, b);
      tmp = *b                   v[i] = '#';              return max + *b;
      *b = a;                    i++;                      }
      return tmp                }
    }                          return i;
  }                          }
}
```

a) Traduce la rutina MaxMin a ensamblador del x86.

```
MaxMin: PUSHL %EBP
        MOVL %ESP,%EBP
        MOVL 12(%EBP),%ECX
        MOVL 8(%EBP),%EDX
        CMPL (%ECX),%EDX
        JLE ELSE
        MOVL %EDX,%EAX
        JMP END
ELSE:   MOVL (%ECX),%EAX
        MOVL %EDX, (%ECX)
END:    MOVL %EBP,%ESP
        POPL %EBP
        RET
```

b) **Traduce** la rutina Matxaca a ensamblador del x86.

```
Matxaca:  PUSHL %EBP
          MOVL %ESP,%EBP
          XORL %EAX,%EAX
          MOVL 8(%EBP),%EDX
WHILE:    CMPB $'.',( %EDX,%EAX)
          JE  END
          MOVB $'#',( %EDX,%EAX)
          INCL %EAX
          JMP WHILE
END:      MOVL %EBP,%ESP
          POPL %EBP
          RET
```

c) **Traduce** la rutina Exa a ensamblador del x86.

```
Exa:  PUSHL %EBP
      MOVL %ESP,%EBP
      PUSHL 12(%EBP)
      PUSHL 8(%EBP)
      CALL MaxMin
      ADDL $8,%ESP
      MOVL 12(%EBP),%EDX
      ADDL (%EDX),%EAX
      MOVL %EBP,%ESP
      POPL %EBP
      RET
```

Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 2. (2 puntos)

Disponemos de un procesador de 32 bits con bus de direcciones de 24 bits y una memoria cache de las siguientes características:

- Directa
- Tamaño total: 4096 bytes
- Tamaño de bloque: 16 bytes
- Política de escritura: copy back + write allocate

Rellena la siguiente tabla, suponiendo que inicialmente la cache está vacía, e indica el contenido final de la cache.

					Lectura de MP			Escritura en MP		
Tipo	@ en hexa	Bloque de MP	Bloque de MC	Acierto /Fallo	si/no	@ hex	tamaño	si/no	@ hex	tamaño
W word	ECA932	ECA93	93	F	SI	ECA930	16	NO		
W long	ECA944	ECA94	94	F	SI	ECA940	16	NO		
R byte	ECC941	ECC94	94	F	SI	ECC940	16	SI	ECA940	16
R byte	ECA932	ECA93	93	A	NO			NO		
W byte	ECA937	ECA93	93	A	NO			NO		
R long	ECC934	ECC93	93	F	SI	ECC930	16	SI	ECA930	16

Contenido final de la cache

Bloque MC	Etiqueta	Dirty bit
93	ECC	
94	ECC	

Problema 3. (2 puntos)

Dado el siguiente código escrito en ensamblador del x86:

```

    movl $0, %ebx
    movl $0, %esi
for:  cmpl $256*1024, %esi
      jge end
(a)   movl (%ebx, %esi, 4), %eax
      shll $2, %eax
(b)   addl %eax, 4*1024(%ebx, %esi, 4)
(c)   addl 12*1024(%ebx, %esi, 4), %eax
      addl $1024, %esi
      jmp for
end:

```

Suponiendo que la memoria virtual utiliza páginas de tamaño 4Ks y que se dispone de un TLB de 3 entradas completamente asociativo con reemplazo LRU, responde a las siguientes preguntas:

- a) Para cada uno de los accesos (etiquetas (a), (b) y (c)), **indica** a qué página de la memoria virtual se accede en cada una de las 16 primeras iteraciones del bucle

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

- b) Indica en cuales de los accesos a memoria ejecutados en las 16 primeras iteraciones son fallo de TLB (F) y cuales son acierto de TLB (A)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a read	F	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
b read	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
b write	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
c read	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

- c) Calcula la cantidad de aciertos de TLB en TODO el bucle

256 iteraciones

Acceso (a): 1 fallo y 255 aciertos

Acceso (b): 256 fallos y 0 aciertos para leer operando en memoria y 256 aciertos para escribir resultado

Acceso (c): 256 fallos y 0 aciertos para leer operando en memoria

Total aciertos: $255 + 0 + 256 + 0 = 511$ aciertos

- d) Calcula la cantidad de fallos de TLB en TODO el bucle

Total fallos: $1 + 256 + 0 + 256 = 513$ fallos

Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 4. (3 puntos)

Tenemos un procesador (que llamaremos procesador original) que funciona a una frecuencia de 2 GHz. Este procesador no tiene ningún tipo de jerarquía de memoria, todos los accesos se realizan directamente sobre memoria principal. En este procesador hemos ejecutado una aplicación A (que usaremos a lo largo de todo el problema) y hemos obtenido los siguientes datos: tiempo de ejecución 18 s, 2×10^9 instrucciones ejecutadas, 3×10^9 accesos a memoria. Sabemos además que cada acceso a memoria tarda 11 ciclos.

a) **Calcula** el CPI del procesador original al ejecutar la aplicación A (a este CPI lo llamaremos CPIoriginal).

$$18 \text{ s} * 2 \times 10^9 \text{ c/s} = 36 \times 10^9 \text{ ciclos}$$
$$\text{CPI}_{\text{original}} = 36 \times 10^9 \text{ c} / 2 \times 10^9 \text{ i} = \mathbf{18 \text{ c/i}}$$

b) **Calcula** el CPI de un procesador (que llamaremos procesador ideal) en que cada acceso tardase 1 ciclo (a este CPI lo denominaremos CPIideal).

$$\text{Penalización/acceso} = 11 \text{ ciclos} - 1 \text{ ciclo} = 10 \text{ ciclos/acceso.}$$
$$\text{Ciclos penalización} = 3 \times 10^9 \text{ accesos} * 10 \text{ ciclos/acceso} = 30 \times 10^9 \text{ ciclos}$$
$$\text{Ciclos ideal} = \text{Ciclos original} - \text{Ciclos penalización} = 36 \times 10^9 \text{ ciclos} - 30 \times 10^9 \text{ ciclos} = 6 \times 10^9 \text{ ciclos}$$
$$\text{CPI}_{\text{ideal}} = \text{Ciclos ideal} / \text{instrucciones} = 6 \times 10^9 \text{ c} / 2 \times 10^9 \text{ i} = \mathbf{3 \text{ c/i}}$$

Para mejorar el rendimiento respecto el procesador original, añadimos una cache 2-asociativa al procesador (que denominaremos procesador 2A). Para poder mantener la frecuencia de 2 GHz, los accesos a la cache 2-asociativa tardan 2 ciclos, es decir que en caso de acierto tenemos una penalización de 1 ciclo. La penalización media en caso de fallo es de 17 ciclos. La aplicación A ejecutada en el procesador 2A tarda 4,7 segundos.

c) **Calcula** la tasa de fallos de la cache 2-asociativa.

$$\text{CPI } 2A = 4,7 \text{ s} * 2 \times 10^9 \text{ c/s} / 2 \times 10^9 \text{ i} = 4,7 \text{ c/i}$$
$$\text{CPI}_{\text{mem}} = \text{CPI } 2A - \text{CPI}_{\text{ideal}} = 4,7 \text{ c/i} - 3 \text{ c/i} = 1,7 \text{ c/i}$$
$$\text{CPI}_{\text{mem}} = nr * ((1-m) * 1 + m * 17) \text{ ----> } (1 - m + 17*m) \text{ ciclos/acceso} = 1,7 \text{ c/i} / 1,5 \text{ a/i} = 1,1333 \text{ ciclos/acceso}$$
$$16 \text{ ciclos/fallo} * m = 0,13333 \text{ ciclos/acceso} \text{ ----> } \mathbf{m = 0,008333 \text{ fallos/acceso}}$$

Uno de los ingenieros ha sugerido el uso de un predictor de vía. El predictor sugerido tendría una tasa de aciertos del 80% y tendría un impacto negligible tanto en área como en la frecuencia y consumo del procesador. Al procesador con cache 2 asociativa y predictor de vía lo denominaremos procesador P. En caso de que el predictor acierte la vía no hay penalización respecto el procesador ideal, si hay fallo de predictor pero acierto de cache se incurre en un ciclo de penalización (tal como ocurría con la cache 2 asociativa), y finalmente, si es fallo de predictor y también de cache, la penalización es de 17 ciclos (también la misma que la cache 2 asociativa).

d) **Calcula** el CPI del procesador P (CPIP).

$$\text{CPI} = \text{CPI ideal} + nr * (\text{mp} * \text{tpp} + m * \text{tpf}) = 3 \text{ c/I} + 1,5 \text{ a/I} * (0,20 \text{ fp/a} * 1 \text{ c/fp} + 0,00833 \text{ fc/a} * 16 \text{ c/fc}) = \mathbf{3,5 \text{ c/i}}$$

e) **Calcula** el speed-up del procesador P respecto el procesador 2a.

$$\begin{aligned} \text{TexeP} &= \text{CPI} * I / F = 3,5 \text{ c/i} * 2 \times 10^9 \text{ i} / 2 \times 10^9 \text{ c/s} = 3,5 \text{ s} \\ \text{Speed-up} &= 4,7 \text{ s} / 3,5 \text{ s} = \mathbf{1,34} \end{aligned}$$

Queremos saber la potencia media debida a conmutación de la jerarquía de memoria del procesador P. Ignoraremos por tanto la potencia disipada por fugas así como la potencia de conmutación del procesador, y también la del predictor que ya hemos comentado que tiene un impacto despreciable. Sabemos que cada vez que se accede una Vía de la cache se consumen 5 nJ (nanojoules) y cada vez que hay un fallo de cache se consumen 20 nJ adicionales.

f) **Calcula** la potencia de conmutación media de la jerarquía de memoria del procesador P al ejecutar la aplicación A

Si hay acierto de predictor solo se accede una vía (5 nJ), si hay fallo de predictor se acceden las 2 vías (10 nJ) y si es fallo de cache 20 nJ adicionales (30 nJ en total)

$$\begin{aligned} \text{Energía} &= 3 \times 10^9 \text{ accesos} * (0,8 * 5 \times 10^{-9} \text{ J} + 0,20 * 10 \times 10^{-9} \text{ J} + 0,0083 * 20 \times 10^{-9} \text{ J}) = 18,5 \text{ Joules} \\ P &= 18,5 \text{ Joules} / 3,5 \text{ s} = \mathbf{5,29 \text{ W}} \end{aligned}$$

Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 1. (3 puntos)

Dado las siguientes rutinas escritas en C:

```
int MaxMin(int a, int *b){      int Matxaca(char v[]){      int Exa(int a, int *b){
    if (a > *b)                  int i;                      int max;
        return a;                i=0;
    else {                      while (v[i] != '\.'){          max = MaxMin(a, b);
        tmp = *b                  v[i] = '#';              return max + *b;
        *b = a;                  i++;                      }
        return tmp                }
    }                            return i;
}                                }
}
```

a) Traduce la rutina MaxMin a ensamblador del x86.

```
MaxMin: PUSHL %EBP
        MOVL %ESP,%EBP
        MOVL 12(%EBP),%ECX
        MOVL 8(%EBP),%EDX
        CMPL (%ECX),%EDX
        JLE ELSE
        MOVL %EDX,%EAX
        JMP END
ELSE:   MOVL (%ECX),%EAX
        MOVL %EDX,(%ECX)
END:    MOVL %EBP,%ESP
        POPL %EBP
        RET
```

b) **Traduce** la rutina Matxaca a ensamblador del x86.

```
Matxaca:  PUSHL %EBP
          MOVL %ESP,%EBP
          XORL %EAX,%EAX
          MOVL 8(%EBP),%EDX
WHILE:    CMPB $'.',( %EDX,%EAX)
          JE  END
          MOVB $'#',( %EDX,%EAX)
          INCL %EAX
          JMP WHILE
END:      MOVL %EBP,%ESP
          POPL %EBP
          RET
```

c) **Traduce** la rutina Exa a ensamblador del x86.

```
Exa:  PUSHL %EBP
      MOVL %ESP,%EBP
      PUSHL 12(%EBP)
      PUSHL 8(%EBP)
      CALL MaxMin
      ADDL $8,%ESP
      MOVL 12(%EBP),%EDX
      ADDL (%EDX),%EAX
      MOVL %EBP,%ESP
      POPL %EBP
      RET
```


Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 2. (2 puntos)

Disponemos de un procesador de 32 bits con bus de direcciones de 24 bits y una memoria cache de las siguientes características:

- Directa
- Tamaño total: 4096 bytes
- Tamaño de bloque: 16 bytes
- Política de escritura: copy back + write allocate

Rellena la siguiente tabla, suponiendo que inicialmente la cache está vacía, e indica el contenido final de la cache.

					Lectura de MP			Escritura en MP		
Tipo	@ en hexa	Bloque de MP	Bloque de MC	Acierto /Fallo	si/no	@ hex	tamaño	si/no	@ hex	tamaño
W word	ECA932	ECA93	93	F	SI	ECA930	16	NO		
W long	ECA944	ECA94	94	F	SI	ECA940	16	NO		
R byte	ECC941	ECC94	94	F	SI	ECC940	16	SI	ECA940	16
R byte	ECA932	ECA93	93	A	NO			NO		
W byte	ECA937	ECA93	93	A	NO			NO		
R long	ECC934	ECC93	93	F	SI	ECC930	16	SI	ECA930	16

Contenido final de la cache

Bloque MC	Etiqueta	Dirty bit
93	ECC	
94	ECC	

Problema 3. (2 puntos)

Dado el siguiente código escrito en ensamblador del x86:

```

    movl $0, %ebx
    movl $0, %esi
for:  cmpl $256*1024, %esi
      jge end
(a)   movl (%ebx, %esi, 4), %eax
      shll $2, %eax
(b)   addl %eax, 4*1024(%ebx, %esi, 4)
(c)   addl 12*1024(%ebx, %esi, 4), %eax
      addl $1024, %esi
      jmp for
end:

```

Suponiendo que la memoria virtual utiliza páginas de tamaño 4Ks y que se dispone de un TLB de 3 entradas completamente asociativo con reemplazo LRU, responde a las siguientes preguntas:

- a) Para cada uno de los accesos (etiquetas (a), (b) y (c)), **indica** a qué página de la memoria virtual se accede en cada una de las 16 primeras iteraciones del bucle

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

- b) Indica en cuales de los accesos a memoria ejecutados en las 16 primeras iteraciones son fallo de TLB (F) y cuales son acierto de TLB (A)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a read	F	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
b read	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
b write	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
c read	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

- c) Calcula la cantidad de aciertos de TLB en TODO el bucle

256 iteraciones

Acceso (a): 1 fallo y 255 aciertos

Acceso (b): 256 fallos y 0 aciertos para leer operando en memoria y 256 aciertos para escribir resultado

Acceso (c): 256 fallos y 0 aciertos para leer operando en memoria

Total aciertos: $255 + 0 + 256 + 0 = 511$ aciertos

- d) Calcula la cantidad de fallos de TLB en TODO el bucle

Total fallos: $1 + 256 + 0 + 256 = 513$ fallos

Cognoms: Nom:

2on Control Arquitectura de Computadors

Curs 2012-2013 Q1

Problema 4. (3 puntos)

Tenemos un procesador (que llamaremos procesador original) que funciona a una frecuencia de 2 GHz. Este procesador no tiene ningún tipo de jerarquía de memoria, todos los accesos se realizan directamente sobre memoria principal. En este procesador hemos ejecutado una aplicación A (que usaremos a lo largo de todo el problema) y hemos obtenido los siguientes datos: tiempo de ejecución 18 s, 2×10^9 instrucciones ejecutadas, 3×10^9 accesos a memoria. Sabemos además que cada acceso a memoria tarda 11 ciclos.

a) **Calcula** el CPI del procesador original al ejecutar la aplicación A (a este CPI lo llamaremos CPIoriginal).

$$18 \text{ s} * 2 \times 10^9 \text{ c/s} = 36 \times 10^9 \text{ ciclos}$$
$$\text{CPI}_{\text{original}} = 36 \times 10^9 \text{ c} / 2 \times 10^9 \text{ i} = \mathbf{18 \text{ c/i}}$$

b) **Calcula** el CPI de un procesador (que llamaremos procesador ideal) en que cada acceso tardase 1 ciclo (a este CPI lo denominaremos CPIideal).

$$\text{Penalización/acceso} = 11 \text{ ciclos} - 1 \text{ ciclo} = 10 \text{ ciclos/acceso.}$$
$$\text{Ciclos penalización} = 3 \times 10^9 \text{ accesos} * 10 \text{ ciclos/acceso} = 30 \times 10^9 \text{ ciclos}$$
$$\text{Ciclos ideal} = \text{Ciclos original} - \text{Ciclos penalización} = 36 \times 10^9 \text{ ciclos} - 30 \times 10^9 \text{ ciclos} = 6 \times 10^9 \text{ ciclos}$$
$$\text{CPI}_{\text{ideal}} = \text{Ciclos ideal} / \text{instrucciones} = 6 \times 10^9 \text{ c} / 2 \times 10^9 \text{ i} = \mathbf{3 \text{ c/i}}$$

Para mejorar el rendimiento respecto el procesador original, añadimos una cache 2-asociativa al procesador (que denominaremos procesador 2A). Para poder mantener la frecuencia de 2 GHz, los accesos a la cache 2-asociativa tardan 2 ciclos, es decir que en caso de acierto tenemos una penalización de 1 ciclo. La penalización media en caso de fallo es de 17 ciclos. La aplicación A ejecutada en el procesador 2A tarda 4,7 segundos.

c) **Calcula** la tasa de fallos de la cache 2-asociativa.

$$\text{CPI } 2A = 4,7 \text{ s} * 2 \times 10^9 \text{ c/s} / 2 \times 10^9 \text{ i} = 4,7 \text{ c/i}$$
$$\text{CPI}_{\text{mem}} = \text{CPI } 2A - \text{CPI}_{\text{ideal}} = 4,7 \text{ c/i} - 3 \text{ c/i} = 1,7 \text{ c/i}$$
$$\text{CPI}_{\text{mem}} = nr * ((1-m) * 1 + m * 17) \text{ ----> } (1 - m + 17*m) \text{ ciclos/acceso} = 1,7 \text{ c/i} / 1,5 \text{ a/i} = 1,1333 \text{ ciclos/acceso}$$
$$16 \text{ ciclos/fallo} * m = 0,13333 \text{ ciclos/acceso} \text{ ----> } \mathbf{m = 0,008333 \text{ fallos/acceso}}$$

Uno de los ingenieros ha sugerido el uso de un predictor de vía. El predictor sugerido tendría una tasa de aciertos del 80% y tendría un impacto negligible tanto en área como en la frecuencia y consumo del procesador. Al procesador con cache 2 asociativa y predictor de vía lo denominaremos procesador P. En caso de que el predictor acierte la vía no hay penalización respecto el procesador ideal, si hay fallo de predictor pero acierto de cache se incurre en un ciclo de penalización (tal como ocurría con la cache 2 asociativa), y finalmente, si es fallo de predictor y también de cache, la penalización es de 17 ciclos (también la misma que la cache 2 asociativa).

d) **Calcula** el CPI del procesador P (CPIP).

$$\text{CPI} = \text{CPI ideal} + nr * (\text{mp} * \text{tpp} + m * \text{tpf}) = 3 \text{ c/I} + 1,5 \text{ a/I} * (0,20 \text{ fp/a} * 1 \text{ c/fp} + 0,00833 \text{ fc/a} * 16 \text{ c/fc}) = \mathbf{3,5 \text{ c/i}}$$

e) **Calcula** el speed-up del procesador P respecto el procesador 2a.

$$\begin{aligned} \text{TexeP} &= \text{CPI} * I / F = 3,5 \text{ c/i} * 2 \times 10^9 \text{ i} / 2 \times 10^9 \text{ c/s} = 3,5 \text{ s} \\ \text{Speed-up} &= 4,7 \text{ s} / 3,5 \text{ s} = \mathbf{1,34} \end{aligned}$$

Queremos saber la potencia media debida a conmutación de la jerarquía de memoria del procesador P. Ignoraremos por tanto la potencia disipada por fugas así como la potencia de conmutación del procesador, y también la del predictor que ya hemos comentado que tiene un impacto despreciable. Sabemos que cada vez que se accede una Vía de la cache se consumen 5 nJ (nanojoules) y cada vez que hay un fallo de cache se consumen 20 nJ adicionales.

f) **Calcula** la potencia de conmutación media de la jerarquía de memoria del procesador P al ejecutar la aplicación A

Si hay acierto de predictor solo se accede una vía (5 nJ), si hay fallo de predictor se acceden las 2 vías (10 nJ) y si es fallo de cache 20 nJ adicionales (30 nJ en total)

$$\begin{aligned} \text{Energía} &= 3 \times 10^9 \text{ accesos} * (0,8 * 5 \times 10^{-9} \text{ J} + 0,20 * 10 \times 10^{-9} \text{ J} + 0,0083 * 20 \times 10^{-9} \text{ J}) = 18,5 \text{ Joules} \\ P &= 18,5 \text{ Joules} / 3,5 \text{ s} = \mathbf{5,29 \text{ W}} \end{aligned}$$