

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.

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

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

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									
W long	ECA944									
R byte	ECC941									
R byte	ECA932									
W byte	ECA937									
R long	ECC934									

Contenido final de la cache

Bloque MC	Etiqueta	Dirty bit

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																
b																
c																

- 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

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

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

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).

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

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.

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).

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

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

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.

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

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

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									
W long	ECA944									
R byte	ECC941									
R byte	ECA932									
W byte	ECA937									
R long	ECC934									

Contenido final de la cache

Bloque MC	Etiqueta	Dirty bit

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																
b																
c																

- 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

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

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

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).

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

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.

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).

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

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