

**Proyecto de Programación en Ensamblador
Estructura de Computadores
Grado en Ingeniería Informática**

Ejemplos de casos de prueba

Departamento de Arquitectura y Tecnología de Sistemas Informáticos

2017-2018 (primer semestre)

Ejemplos

A continuación se incluye varios ejemplos de casos de prueba con los argumentos que se pasan a las distintas subrutinas y las direcciones de memoria que se modifican.

Estos ejemplos no constituyen un juego de ensayo completo. No sirven por sí solos para comprobar el correcto funcionamiento de las subrutinas ni para detectar en qué situaciones presentan comportamientos erróneos. Sin embargo, se pueden seguir como guía para la elaboración de juegos de ensayo más completos.

Todos los ejemplos se han descrito utilizando el formato de salida que ofrece el simulador del 88110. En este procesador el direccionamiento se hace a nivel de byte y se utiliza el formato *little-endian*. En consecuencia, cada una de las palabras representadas a continuación de la especificación de la dirección debe interpretarse como formada por 4 bytes con el orden que se muestra en el ejemplo siguiente:

Direcciones de memoria, tal como las representa el simulador:

60000	04050607	05010000
-------	----------	----------

Direcciones de memoria, tal como se deben interpretar:

60000	04
60001	05
60002	06
60003	07

60004	05
60005	01
60006	00
60007	00

Valor de las palabras almacenadas en las posiciones 60000 y 60004, tal como la interpreta el procesador:

60000	0x07060504 = 117.835.012
60004	0x00000105 = 261

Número de filtrados

Caso 1. Llamada a nFiltrados

Llama a 'nFiltrados', pasándole un parámetro no nulo para incrementar la variable nF, que tiene un valor positivo

r30=36848 (0x8FFC)

Direcciones de memoria:

00000 05000000

36848

01000000

Resultado:

r30=36848 (0x8FFC) r29=6 (0x6)

Direcciones de memoria:

00000 06000000

Caso 2. Llamada a nFiltrados

Llama a 'nFiltrados', pasándole un parámetro no nulo para incrementar la variable nF, que tiene un valor cercano al máximo

r30=36848 (0x8FFC)

Direcciones de memoria:

00000 0B000000

36848

FFFFFFFF

Resultado:

r30=36848 (0x8FFC) r29=-1 (0xFFFFFFFF)

Direcciones de memoria:

00000 0C000000

Compara dos imágenes

Caso 3. Llamada a Comp

Llama a 'Comp', pasándole dos imágenes de 4x8 elementos que difieren en uno solo de ellos

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	28800000
32768	04000000	08000000	00000000	00000000
32784	00000000	00070000	00000000	00000000
32800	00000000	00000000		
32800			04000000	08000000
32816	00000000	00000000	00000000	00000000
32832	00000000	00000000	00000000	00000000

Resultado:

r30=36856 (0x8FF8) r29=7 (0x7)

Caso 4. Llamada a Comp

Llama a 'Comp', pasándole dos imágenes de 4x8 elementos en las que difieren todos sus elementos en una unidad

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	28800000
32768	04000000	08000000	55FF55FF	55FF55FF
32784	FF55FF55	FF55FF55	55FF55FF	55FF55FF
32800	FF55FF55	FF55FF55		
32800			04000000	08000000
32816	54FE54FE	54FE54FE	FE54FE54	FE54FE54
32832	54FE54FE	54FE54FE	FE54FE54	FE54FE54

Resultado:

r30=36856 (0x8FF8) r29=32 (0x20)

Extracción de submatriz

Caso 5. Llamada a SubMatriz

Llama a 'SubMatriz', pasándole una imagen de 3x3 elementos de la que se ha de extraer la subimagen correspondiente al elemento central.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	40800000	01000000	01000000
32768	03000000	03000000	10203040	50607080
32784	90000000			
32832	FFFFFFFF	FFFFFFFF	FFFFFFFF	

Resultado:

r30=36848 (0x8FF0)

Direcciones de memoria:

32832	10203040	50607080	90FFFFFF
-------	----------	----------	----------

Caso 6. Llamada a SubMatriz

Llama a 'SubMatriz', pasándole una imagen de 5x5 elementos de la que se ha de extraer la subimagen correspondiente a una esquina (inferior derecha).

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	40800000	04000000	04000000
32768	05000000	05000000	01020304	05060708
32784	090A0B0C	0D0E0F10	11121314	15161718
32800	19000000			
32832	FFFFFFFF	FFFFFFFF	FFFFFFFF	

Resultado:

r30=36848 (0x8FF0)

Direcciones de memoria:

32832	19191919	19191919	19FFFFFF
-------	----------	----------	----------

Valor del píxel filtrado

Caso 7. Llamada a ValorPixel

Llama a 'ValorPixel', pasándole una subimagen nula excepto en su elemento central y un filtro identidad.

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	10800000
32768	00000000	55000000	00000000	
32784	00000000	00000000	00000000	00000000
32800	01000000	00000000	00000000	00000000
32816	00000000			

Resultado:

r30=36856 (0x8FF8) r29=85 (0x55)

Caso 8. Llamada a ValorPixel

Llama a 'ValorPixel', pasándole una subimagen no nula y un filtro que devuelve la suma de los ocho elementos que le rodean.

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	10800000
32768	10111213	14151617	18000000	
32784	01000000	01000000	01000000	01000000
32800	00000000	01000000	01000000	01000000
32816	01000000			

Resultado:

r30=36856 (0x8FF8) r29=160 (0xA0)

Caso 9. Llamada a ValorPixel

Llama a 'ValorPixel', pasándole una subimagen no nula y un filtro que dobla y cambia el signo del elemento al que se aplica.

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	10800000
32768	00000000	55000000	00000000	
32784	00000000	00000000	00000000	00000000
32800	FFFFFFFF	00000000	00000000	00000000
32816	00000000			

Resultado:

r30=36856 (0x8FF8) r29=-170 (0xFFFFFFFF56)

Filtro de un píxel**Caso 10. Llamada a FilPixel**

Llama a 'FilPixel', pasándole una imagen de 4x8 elementos y un filtro identidad que se aplica al único elemento no nulo.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	01000000	01000000	30800000
32768	04000000	08000000	00000000	00000000
32784	00330000	00000000	00000000	00000000
32800	00000000	00000000		
32816	00000000	00000000	00000000	00000000
32832	01000000	00000000	00000000	00000000
32848	00000000			

Resultado:

r30=36848 (0x8FF0) r29=51 (0x33)

Caso 11. Llamada a FilPixel

Llama a 'FilPixel', pasándole una imagen de 4x8 elementos y un filtro que devuelve la media de los ocho elementos que le rodean.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	02000000	02000000	30800000
32768	04000000	08000000	44444444	44444444
32784	44343433	44444444	44448844	44444444
32800	44444444	44444444		
32816	01000000	01000000	01000000	01000000
32832	00000000	01000000	01000000	01000000
32848	01000000			

Resultado:

r30=36848 (0x8FF0) r29=61 (0x3D)

Caso 12. Llamada a FilPixel

Llama a 'FilPixel', pasándole una imagen de 4x8 elementos y un filtro que devuelve una media de los ocho elementos que le rodean menos un factor que multiplica al pixel central. El resultado se ajusta al valor máximo (255). Los registros parten de valores distintos de 0.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	02000000	02000000	28800000
32768	04000000	08000000	43424140	47464544
32784	4B4A4948	4F4E4D4C	43420040	47464544
32800	4B4A4948	4F4E4D4C		
32800			01000000	01000000
32816	01000000	01000000	FFFFFFF8	01000000
32832	01000000	01000000		

Resultado:

r30=36848 (0x8FF0) r29=255 (0xFF)

Filtro de imagen

Caso 13. Llamada a Filtro

Llama a 'Filtro'pasándole una imagen no nula de 4x8 elementos y un filtro que multiplica por 4 cada elemento y le resta tres veces el valor del situado en la misma columna de la fila anterior. Algunos elementos alcanzan el valor máximo.

r30=36852 (0x8FF4)

Direcciones de memoria:

36848		00800000	28800000	50800000
32768	04000000	08000000	01020304	04050607
32784	11421314	14851617	21222324	74252627
32800	31323334	34353637		
32800			00000000	00000000
32816	00000000	00000000	00000000	00000000
32832	00000000	00000000	00000000	00000000
32848	00000000	FDFFFFFF	00000000	00000000
32864	04000000	00000000	00000000	00000000
32880	00000000			

Resultado:

r30=36852 (0x8FF4)

Direcciones de memoria:

32800			04000000	08000000
32816	04080C10	1014181C	44FF4C50	50FF585C
32832	84888C90	FF94989C	C4C8CCD0	D0D4D8DC

Caso 14. Llamada a Filtro

Llama a 'Filtro'pasándole una imagen de 4x6 elementos y un filtro que sustituye cada píxel por la media de los que están situados en los cuatro vértices de la submatriz que lo rodea.

r30=36852 (0x8FF4)

Direcciones de memoria:

36848		00800000	20800000	40800000
32768	04000000	06000000	01020304	05060002
32784	04010305	0306090C	0F120408	10204080
32800	A5A5A5A5	A5A5A5A5	A5A5A5A5	A5A5A5A5
32816	A5A5A5A5	A5A5A5A5	A5A5A5A5	A5A5A5A5
32832	01000000	00000000	01000000	00000000
32848	00000000	00000000	01000000	00000000
32864	01000000			

Resultado:

r30=36852 (0x8FF4)

Direcciones de memoria:

32800	04000000	06000000	01020304	05060004
32816	06080A05	03060A15	29120408	10204080

Caso 15. Llamada a Filtro

Llama a 'Filtro'pasándole una imagen de 4x6 elementos y un filtro identidad formado con un coeficiente negativo.

r30=36852 (0x8FF4)

Direcciones de memoria:

36848		00800000	20800000	40800000
32768	04000000	06000000	00000000	00000020
32784	10300000	00000000	00000000	00000000
32800	A5A5A5A5	A5A5A5A5	A5A5A5A5	A5A5A5A5
32816	A5A5A5A5	A5A5A5A5	A5A5A5A5	A5A5A5A5
32832	00000000	00000000	00000000	00000000
32848	FFFFFFFF	00000000	00000000	00000000
32864	00000000			

Resultado:

r30=36852 (0x8FF4)

Direcciones de memoria:

32800	04000000	06000000	00000000	00000020
32816	10300000	00000000	00000000	00000000

Filtro recursivo de una imagen

Caso 16. Llamada a FiltRec

Llama a 'FiltRec'pasándole una imagen no nula de 4x8 elementos y un filtro que sustituye cada elemento por la media de los que le rodean. El parámetro NCambios tiene valor 2.

r30=36848 (0x8FF0)

Direcciones de memoria:

00000	00000000			
36848	00800000	54800000	30800000	02000000
32768	04000000	08000000	00FF00FF	FF00FF00
32784	00FF00FF	FF00FF00	00FF00FF	FF00FF00
32800	00FF00FF	FF00FF00		
32848		00000000	00000000	00000000
32864	00000000	00000000	00000000	00000000
32880	00000000	00000000	00000000	
32816	01000000	01000000	01000000	01000000
32832	00000000	01000000	01000000	01000000
32848	01000000			

Resultado:

r30=36848 (0x8FF0) r29=0 (0x0)

Direcciones de memoria

00000	08000000			
32768	04000000	08000000	00FF00FF	FF00FF00
32784	00FF00FF	FF00FF00	00FF00FF	FF00FF00
32800	00FF00FF	FF00FF00		
32848		04000000	08000000	00FF00FF
32864	FF00FF00	004C8B9D	9D8B4C00	004C8B9D
32880	9D8B4C00	00FF00FF	FF00FF00	

Caso 17. Llamada a FiltRec

Llama a 'FiltRec'pasándole una imagen no nula de 4x8 elementos y un filtro que sustituye cada elemento por la media de los que le rodean. El parámetro NCambios tiene valor 0, por lo que se detiene al alcanzar el máximo número de llamadas recursivas.

r30=36848 (0x8FF0)

Direcciones de memoria:

00000 00000000

36848 00800000 54800000 30800000 00000000

32768 04000000 08000000 FF0000FF FF0000FF

32784 FF0000FF FF0000FF FF0000FF FF0000FF

32800 FF0000FF FF0000FF

32848 00000000 00000000 00000000 00000000

32864 00000000 00000000 00000000 00000000

32880 00000000 00000000 00000000

32816 01000000 01000000 01000000 01000000

32832 00000000 01000000 01000000 01000000

32848 01000000

Resultado:

r30=36848 (0x8FF0) r29=0 (0x00)

Direcciones de memoria

00000 0C000000

36848 00800000 30800000 54800000 00000000

32768 04000000 08000000 FF0000FF FF0000FF

32784 FF0000FF FF0000FF FF0000FF FF0000FF

32800 FF0000FF FF0000FF

32848 04000000 08000000 FF0000FF

32864 FF0000FF FF8F7795 95778FFF FF8F7795

32880 95778FFF FF0000FF FF0000FF

Caso 18. Llamada a FiltRec

Llama a 'FiltRec'pasándole una imagen no nula de 4x8 elementos y un filtro que sustituye cada cada píxel por la media de los que están situados en los cuatro vértices de la submatriz que lo rodea. El parámetro NCambios tiene valor 1.

r30=36848 (0x8FF0)

Direcciones de memoria:

00000	00000000			
36848	00800000	54800000	30800000	01000000
32768	04000000	08000000	FF0000FF	FF0000FF
32784	FF0000FF	FF0000FF	FF0000FF	FF0000FF
32800	FF0000FF	FF0000FF		
32848		00000000	00000000	00000000
32864	00000000	00000000	00000000	00000000
32880	00000000	00000000	00000000	
32816	01000000	00000000	01000000	00000000
32832	00000000	00000000	01000000	00000000
32848	01000000			

Resultado:

r30=36848 (0x8FF0) r29=0 (0x00)

Direcciones de memoria

00000	06000000			
32768	04000000	08000000	FF0000FF	FF0000FF
32784	FF0000FF	FF0000FF	FF0000FF	FF0000FF
32800	FF0000FF	FF0000FF		
32848		04000000	08000000	FF0000FF
32864	FF0000FF	FFA18882	8288A1FF	FFA18882
32880	8288A1FF	FF0000FF	FF0000FF	

16 Proyecto de Programación en Ensamblador. Ejemplos de casos de prueba (5/10/2017)

```
;      Ejemplo de programa de prueba de la subrutina COMPARA

LEA:    MACRO    (ra, eti)
        or ra, r0, low(eti)
        or.u ra, ra, high(eti)
        ENDMACRO

PUSH:   MACRO(ra)
        subu r30, r30, 4
        st ra, r30, 0
        ENDMACRO

POP:    MACRO(ra)
        ld ra, r30, 0
        addu r30, r30, 4
        ENDMACRO

; Definición de la pila
        org      0xF000
PILA:   data      0

; Definición de las imágenes de prueba (2x2)
        org      0x1000
IMG1:   data      3, 3, 0x07050301, 0x080A0C0F, 0x00000011

        org      0x2000
IMG2:   data      3, 3, 0x07040301, 0x090A0C0F, 0x00000011

; Programa principal
        org      0
PPAL:   LEA (r30, PILA)      ; Inicialización del puntero de pila
        LEA (r1, IMG1)      ; r1: dirección de comienzo de IMG1
        LEA (r2, IMG2)      ; r2: dirección de comienzo de IMG2

        PUSH (r2)           ; Paso de parámetro Imagen2
        PUSH (r1)           ; Paso de parámetro Imagen1

        bsr COMPARA         ; Llamada a la subrutina COMPARA
                                ; r29 debe valer 2 (0x000000002)

        POP (r1)
        POP (r2)
        stop

COMPARA: PUSH (r1)
;      ....
```