3 de Septiembre del 2020

Guía 1
Data ALU

Integrantes:

- Juan Manuel Romaris, Legajo:57108
- Lucero Guadalupe Fernandez Legajo:57485
- Ezequiel Vijande, Legajo:58057

Ejercicio 1

```
Si inicialmente los registros se encuentran en este estado:
a     = $FFFFFFFFFFFFFF
b     = $FFFFFFFFFFFFFF
x     = $FFFFFFFFFFFF

y se ejecuta el siguiente código:
move  #$3D, x1
move  #$3D, a1
move  #$3D, b

Indicar el estado final de los registros
```

a = FF | 00003D | FFFFFF
b = 00 | 00003D | FFFFFF
x = 00003D | FFFFFF

Ejercicio 2

```
Si inicialmente los registros se encuentran en este estado:
a     = $00000000000000
b     = $00000000000000
x     = $000000000000

y se ejecuta el siguiente código:
move  #$CABA00, x1
move  x1, a
move  x1, b1

Indicar el estado final de los registros
```

a = FF | CABA00| 000000
b = 00 | CABA00| 000000
x = CABA00| 000000

Ejercicio 3

```
Si inicialmente los registros se encuentran en este estado:
a     = $00 | a00000 | 000000
y     = $000000 | 000000
ccr   = $00

y se ejecuta el siguiente código:
move  a1, x1
move  a, y1
move  a, r7
move  a1, x0

Indicar el estado final de los registros
```

x = a00000 | a00000
y = a00000 | **000000**
r7 = $0000
ccr = b 0000 0000

Ejercicio 4

Si inicialmente los registros se encuentran en este estado:

```
a     = $00000123800000
b     = $FF000000FFFFFF
x     = $400000400000
```

y se ejecuta el siguiente código:

```
macr  x0,x1,a
rnd   b
mpyr  x0,x1,b
```

Indicar el estado final de los registros

a = 00 | 200124| 000000
b = 00 | 200000| 000000

Ejercicio 5

Si inicialmente los registros se encuentran en este estado:

```
a     = $00000000000000
sr    = $0300
```

y se ejecuta el siguiente código:

```
move  $400000,x0
add   x0,a
add   x0,a
```

1) indicar el estado final de los registros y justificar este resultado
   a = **00 | 800000 | 000000**
   sr = **$0320**
   Se prende el bit E (extensión) porque indica que se trata de un número entero + una fracción.

2) repetir considerando que inicialmente sr = $0700
   a = **00 | 800000 | 000000**
   sr = **$0700**
   El bit E de extension no se prende al estar activado el bit S0 (scaling mode) del SR, pues este bit es un scale down (1 bit arithmetic right shift). Entonces ahora el valor de A es fracción pura.

Ejercicio 6

Si inicialmente los registros se encuentran en este estado:

```
a     = $00000000000000
x     = $0c0000600000
r0    = $0000
```

y se ejecuta el siguiente código:

```
add   x1,a
rep   #$a
norm  r0,a
```

```
add   x0,a
```

Indicar el estado final de los registros

a   = $00C00000000000
r0   = $FFFD

Además, indicar los cambios que se producen en el CCR a lo largo de la ejecución:
Cuando se ejecuta la instrucción **add x1,a** se prende el bit U del CCR. En el tercer llamado de **norm r0,a** este bit se apaga. Luego al realizar la suma **add x0,a** se prenden ambos bits U y E del CCR.

Ejercicio 7

Si inicialmente los registros se encuentran en este estado:

```
r0    = $0000    m0 = $ffff
r4    = $0000    m4 = $ffff
sr    = $0800
```

Se tiene el siguiente mapa de memoria:

```
X:$0000    $10fedc
X:$0001    $210fed
X:$0002    $4210fe
X:$0003    $84210f
X:$0004    $d84210
X:$0005    $fb8421
```

y se ejecuta el siguiente código:

```
      move  x:(r0)+,a
      rep   #6
      move a,y:(r4)+   x:(r0)+,a
      jlc OK
      bset #0,y:$100
OK    bclr #6,sr
```

Indicar el estado final de la memoria Y.
¿Qué significado tiene la memoria Y:$100?

```
y:$0000=        $21fdb8        $421fda        $7fffff        $800000
y:$0004=        $b08420        $f70842        $000000        $000000
y:$0008=        $000000        $000000        $000000        $000000
```

En la operación bset #0,y:$100 se modifica el bit menos significativo de la posición de memoria $0100 de la memoria Y. Justamente esto ocurre dado que y:$100 se refiere a la posición y:$0100.

Ejercicio 8

Escribir la subrutina vect_max.
Compara elemento a elemento los vectores A y B, guarda el valor con modulo mayor en B. Recibe la dirección de inicio de los vectores en r0 y r4, y la cantidad de elementos en n0.

Utilizar la instrucción LOOP, y optimizar la cantidad de instrucciones dentro del bucle a la minima posible. Hint: considerar las instrucciones Tcc (transfer condicional).

Escribir un main de prueba y simular el resultado.

Respuesta:
El código se encuentra en un archivo aparte dentro de la carpeta de entrega con el nombre de 'Ej8.asm'.