



Res. 2333 del 2012

Vigilada Mineducación Resolución 12220 de 2016

ARQUITECTURA DE COMPUTADORES II: LABORATORIO.

Session 3 – August 10

Session 3: Outline

- Directives
- Memory Map
- STR Instruction



Res. 2333 del 2012

Vigilada Mineducación Resolución 12220 de 2016

DIRECTIVA EQU



- Define una constante o una dirección fija. Asocia un numero a una etiqueta.

COUNT EQU 0x25

- VENTAJA: Cambiar valores en muchas partes del programa

DATA1 EQU 0x39 ;the way to define hex value

DATA2 EQU 2_00110101 ;the way to define binary value (35 in hex)

DATA3 EQU 39 ;decimal numbers (27 in hex)

DATA4 EQU '2' ;ASCII characters

DIRECTIVA RN



- Se usa para definir el nombre de un registro,

```
1      AREA PRIMER_EJEMPLO, CODE, READONLY
2 COUNT EQU 0x25
3 VAL1  RN  R1
4      ENTRY
5          LDR VAL1, =0x1100
6          MOV R2, #0x34
7          MOV R3, #COUNT
8          ADD R3, R2, R1
9          STR R3, [R1]
10 HERE  B    HERE
11      END
```

INSTRUCCIÓN ARM.

```
1      AREA SUMA_SIMPLE, CODE, READONLY
2      ARM
3      ENTRY
4      MOV R0, #0X25
5      ADD R2, R0, #0X34
6  HERE      B      HERE
7  END
```

CPSR	0x000000D3
N	0
Z	0
C	0
V	0
I	1
F	1
T	0
M	0x13

MAPA DE MEMORIA



Res. 2333 del 2012

Vigilada Mineducación Resolución 12220 de 2016

```
1      AREA SUMA_SIMPLE, CODE, READONLY
2      ARM
3  AD_RAM EQU 0XE0001000
4  RAM    RN  R11
5      ENTRY
6      LDR RAM, =AD_RAM
```

Current Mapped

000: 0x00000000 - 0x0000000B exec read
001: 0x7FFFFFF0 - 0x7FFFFFF3 exec read write
002: 0x7FFFFFF4 - 0x7FFFFFFF read write
003: 0xE0000000 - 0xE0017FFF read write
004: 0xE001C000 - 0xE002FFFF read write
005: 0xE01FC000 - 0xE01FFFFF read write
006: 0xFFFFF000 - 0xFFFFFFFF read write

Kill Selected Range

Map Range - Example: 0x40000000, 0x4000FFFF

- ☐ Read
☐ Write
☐ Execute

Map Range

STR Instruction



Res. 2333 del 2012

Vigilada Mineducación Resolución 12220 de 2016

Is used to transfer contents of a register to memory.

STR	Store 32 bits (word)
STRB	Store 1 byte
STRH	Store 16 bits (Half word)
STM	Store multiple words

```
1          AREA SUMA_SIMPLE, CODE, READONLY
2          ARM
3 AD_RAM EQU 0XE0001000
4 RAM    RN  R11
5          ENTRY
6          LDR RAM, =AD_RAM
7          MOV R0,    #0X25
8          ADD R2, R0, #0X34
9          STR R2, [RAM]
10 HERE   B    HERE
11          END
```

```
1      AREA SUMA_SIMPLE, CODE, READONLY
2      ARM
3  AD_RAM EQU 0XE0001000
4  RAM    RN  R11
5
6      ENTRY
7
8      LDR RAM, =AD_RAM
9      MOV R0,    #0X25
10     ADD R2, R0, #0X34
11     STRB R2, [RAM]
12
13     HERE B     HERE
14
15     END
```

enguajeMaquina_2018II\\Keil_Source\\Suma_Simple\\Objects\\Suma_Simple.axf

Memory 1

Address: 0XE0001000

0xE0001000: 59 00 00 00


```

1      AREA SUMA_SIMPLE, CODE, READONLY
2      ARM
3  AD_RAM EQU 0XE0001000
4  RAM    RN  R11
5
6      ENTRY
7
8      LDR RAM, =AD_RAM
9      MOV R0,    #0X25
10     ADD R2, R0, #0X34
11     STRH R2, [RAM]
12
13     HERE B     HERE
14
15     END

```

nguajeMaquina_2018II\\Keil_Source\\Suma_Simple\\Objects\\Suma_Simple.axf



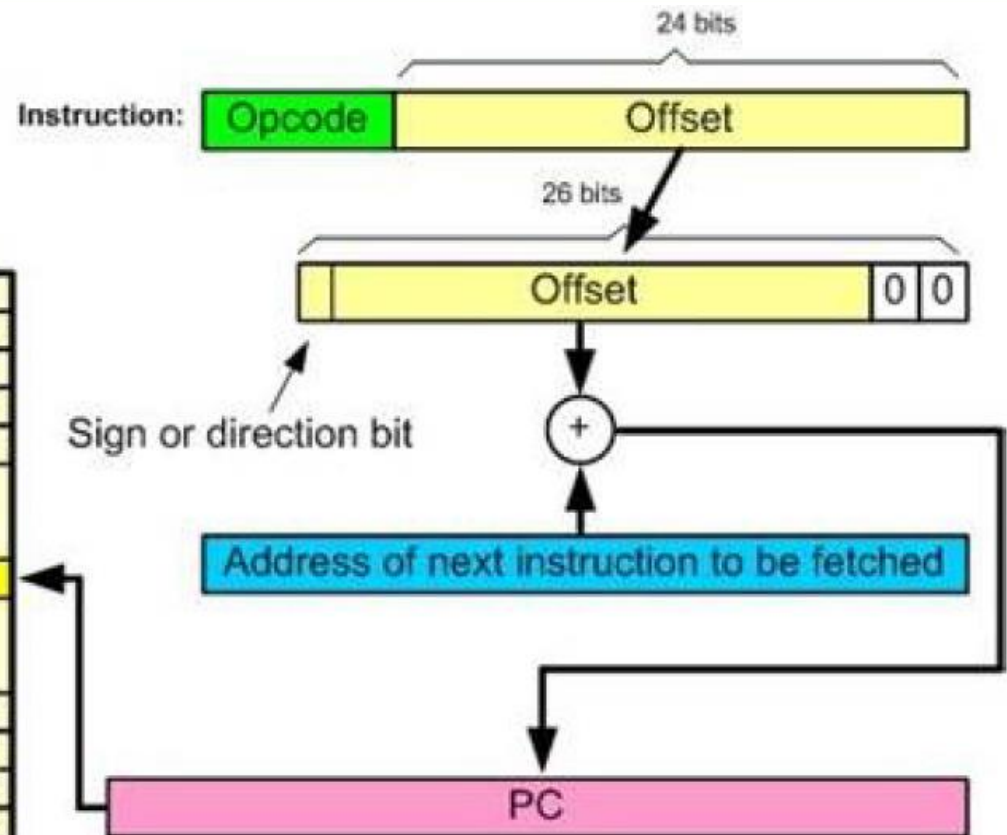
Memory 1

Address: 0XE0001000

0xE0001000: 59 00 00 00

0x00000003	0x00000002	0x00000001	0x00000000
0x00000007	0x00000006	0x00000005	0x00000004
0x0000000B	0x0000000A	0x00000009	0x00000008
0x0000000F	0x0000000E	0x0000000D	0x0000000C
0x00000013	0x00000012	0x00000011	0x00000010
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
0xFFFFFFFF3	0xFFFFFFFF2	0xFFFFFFFF1	0xFFFFFFFF0
0xFFFFFFFF7	0xFFFFFFFF6	0xFFFFFFFF5	0xFFFFFFFF4
0xFFFFFFFFB	0xFFFFFFFFA	0xFFFFFFFF9	0xFFFFFFFF8
0xFFFFFFFFF	0xFFFFFFF	0xFFFFFFF	0xFFFFFFF

Program memory (Addresses are word-aligned)



Máximo 32M Instrucciones en cada salto de memoria

EJERCICIO.



- Guarde en memoria las palabras 0x01100050, 0x14231823 y 0x10111232; luego acceda a los bytes de tal forma que pueda realizar las siguientes operaciones teniendo en cuenta que $A=0x23$, $B=0x32$ y $C=0x50$.
- $[(A + B) + C](0xA5)$ y $[(A + B) - (A - C)]0x82$
- Realice las tareas solicitadas en el menor número de líneas de ASM ARM posibles.
- No debe cargar los operandos directamente.