

SIMULADOR DSP

LABORATORIO DE DSP-FPGA

Agosto 2023

Instituto Tecnológico de Buenos aires

Introducción

Contenido de la Clase:

- Repaso Arquitectura DSP
 - ALU
 - PCU
 - AGU
- Simulador
 - Funciones de Carga Flujo y Debug de programa
 - Ejercicios
 - Carga de Archivos (breve CLA, LOAD)
 - Presentación TP1

Arquitectura DSP

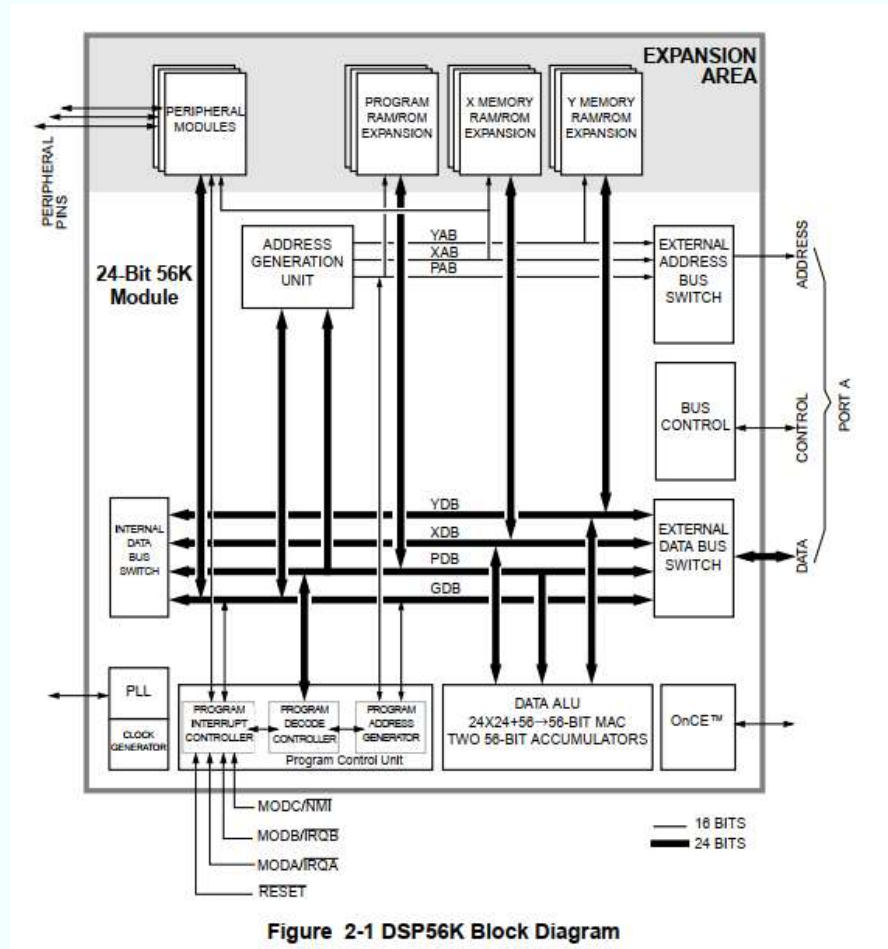
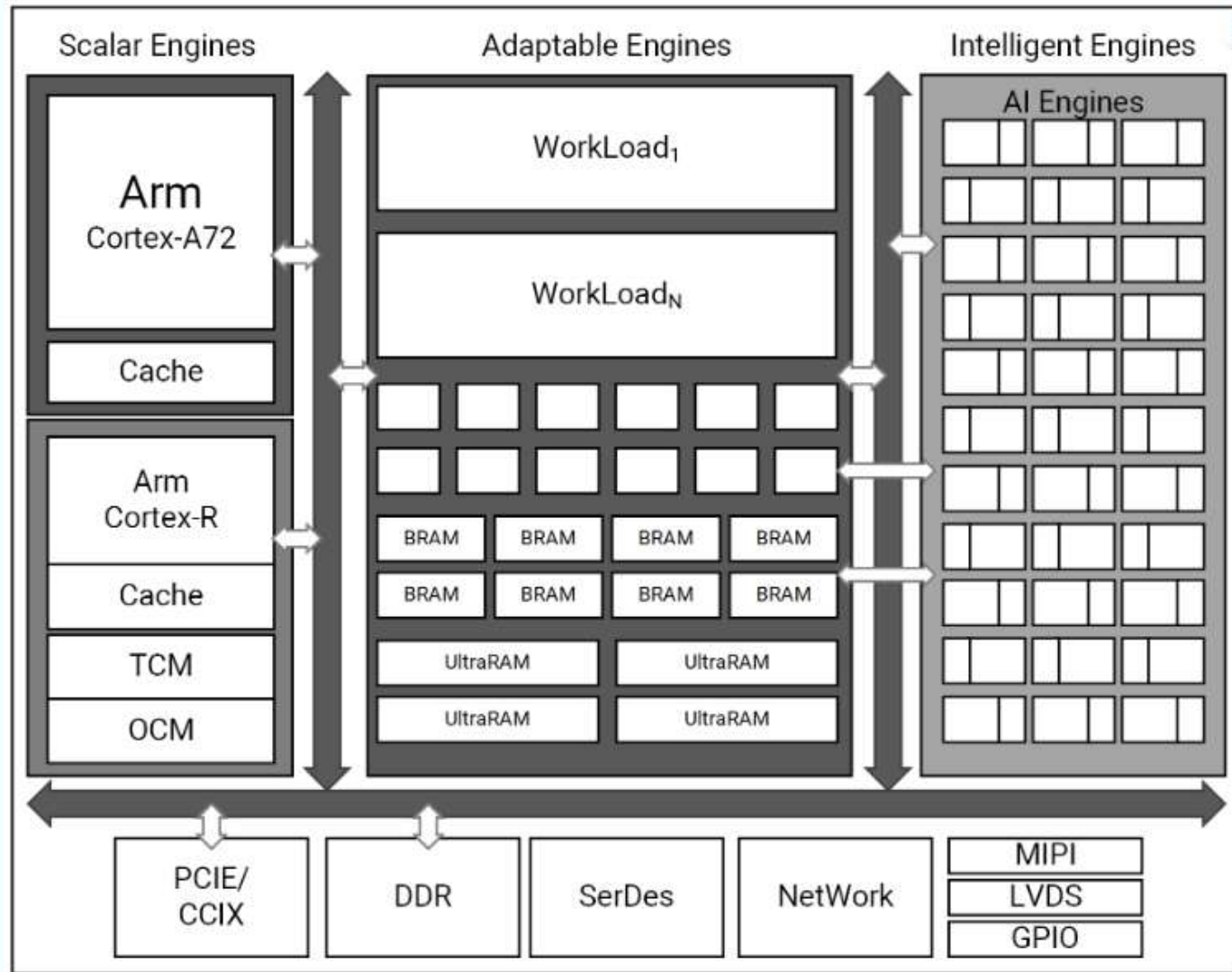
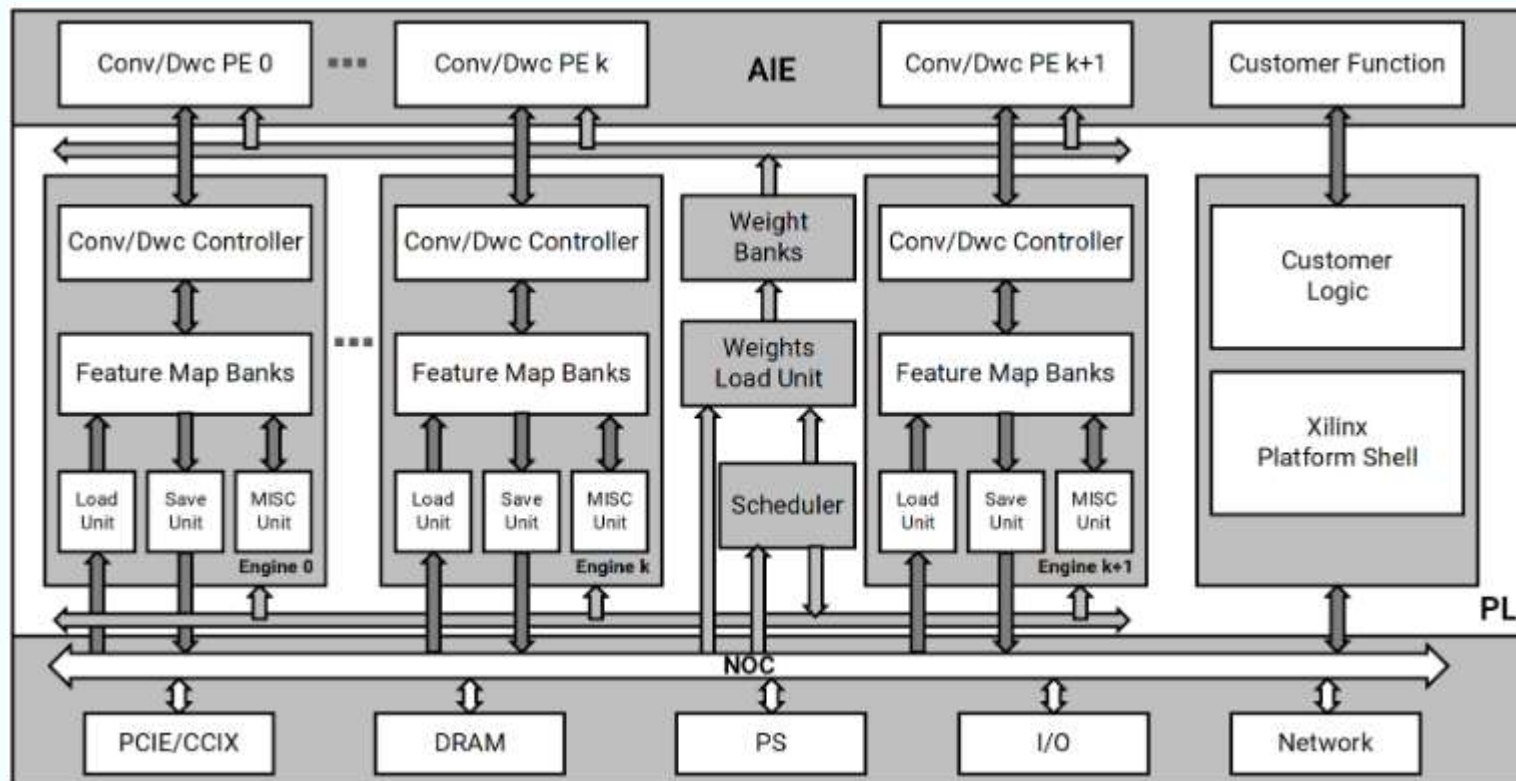


Figure 2-1 DSP56K Block Diagram

Arquitectura FPGA con HPS y AIE



Arquitectura FPGA con HPS y AIE



Program Controller Unit

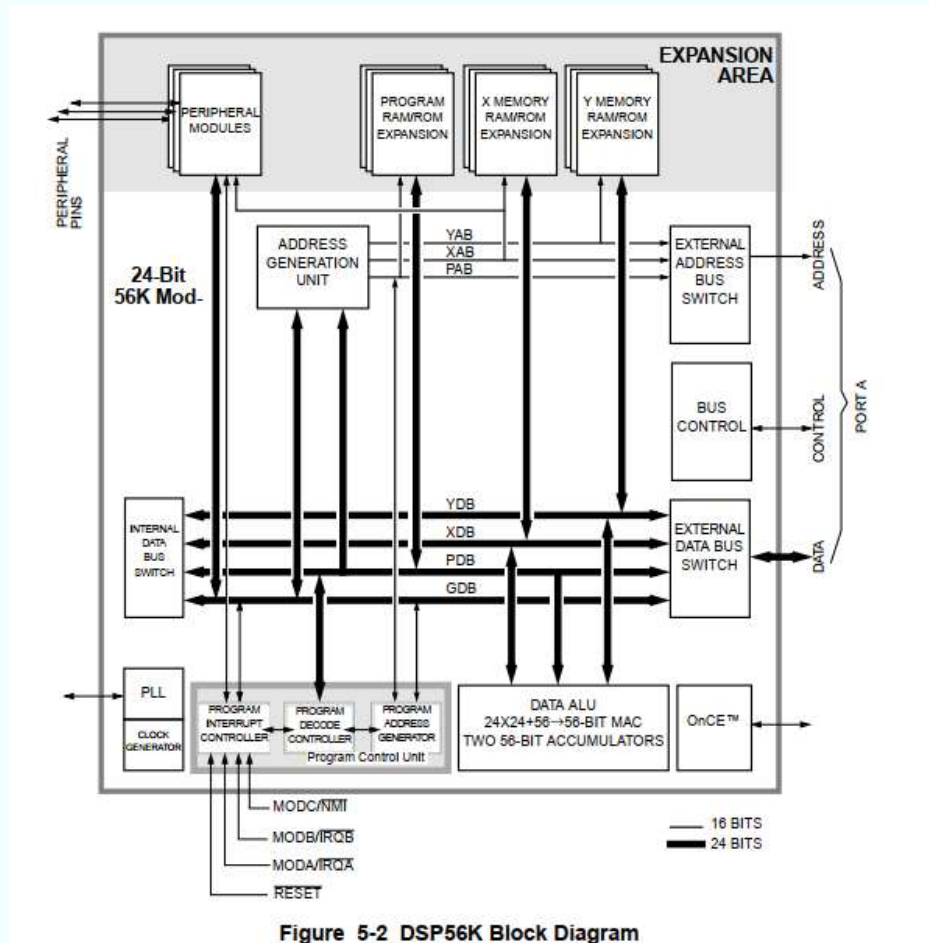


Figure 5-2 DSP56K Block Diagram

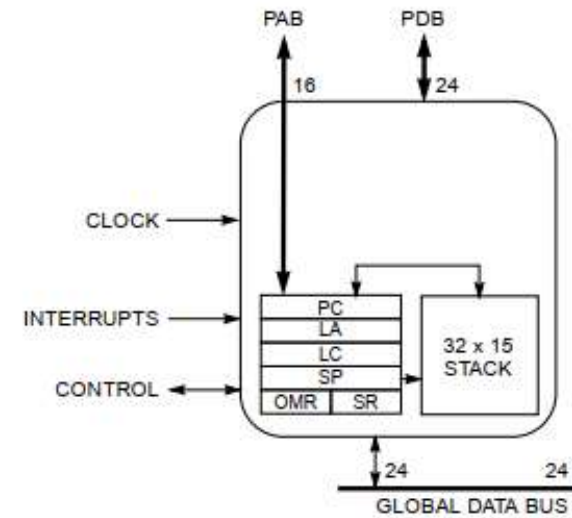


Figure 5-1 Program Address Generator

Address Generator Unit

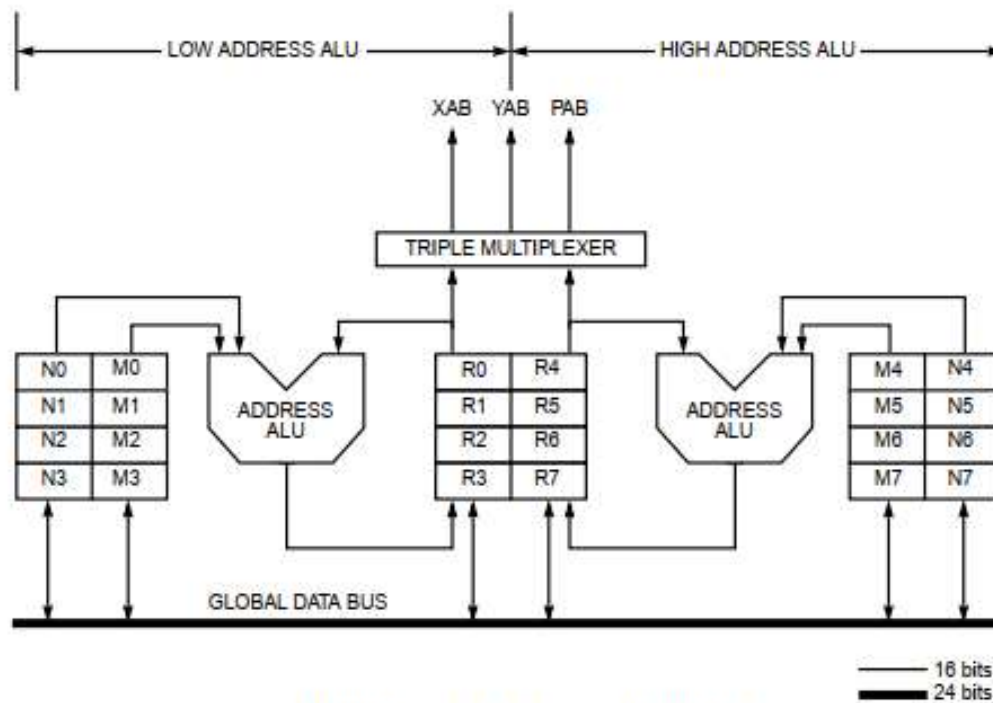


Figure 4-2 AGU Block Diagram

DATA ALU

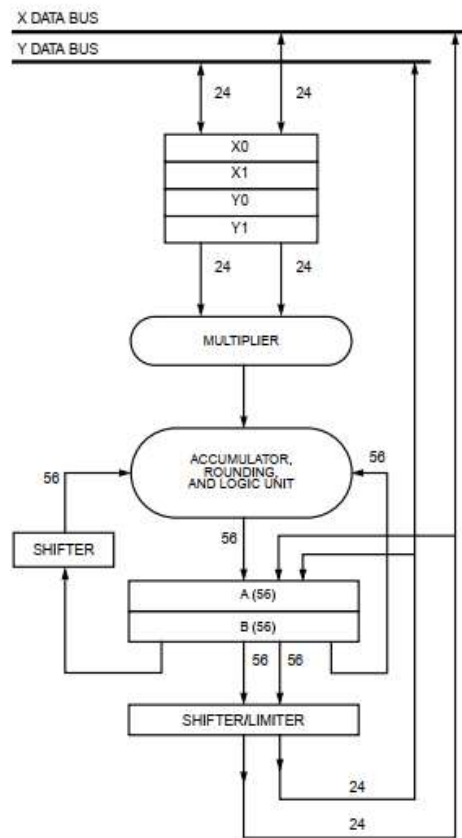


Figure 3-2 Data ALU

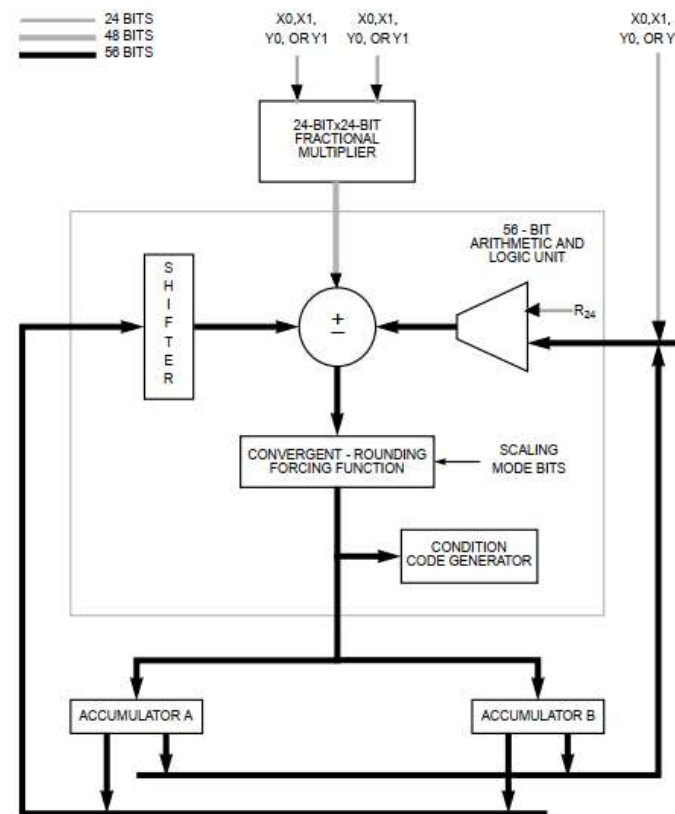


Figure 3-3 MAC Unit

Comandos del Simulador

Display:

- Display ON p:\$e000..p:\$e00f
 - La posición de memoria \$e003 se mostrara cada vez ingresemos el comando “display”
- Display off X
 - Deja de mostrar el registro X en el display
- Display R [W,RW] x:\$1000..\$100f
 - Muestra el bloque de memoria solo si hubo un acceso para lectura o escritura después del ultimo display.

Comandos del Simulador

ASM:

- ASM
 - Se puede empezar a ingresar instrucciones de assembler desde la ubicación del PC
- ASM p:\$e000
 - Inicia la carga de instrucciones de assembler desde la ubicación de memoria indicada.
- ASM p:\$e000 move x1,x:(R0)+
 - Ingresa una única instrucción en memoria

Comandos del Simulador

CHANGE:

- Change PC \$e000
 - Setea el registro en un valor determinado
- Change x:\$0000..\$0004 %1100
 - Carga directa de valores en memoria

DISASSEMBLE:

- DISASSEMBLE p:\$e000..\$e00f
 - Muestra el contenido de memoria e instrucciones que representan.

Comandos del Simulador

TRACE:

- trace #20 h
 - Ejecuta las próximas 20 instrucciones, se detiene si hay un breakpoint.

GO:

- Go \$e003 #1 :3
 - Inicia la simulación en modo continuo, desde PC=\$e003, se detiene en la 3° ocurrencia del breakpoint “1”.

Comandos del Simulador

BREAK:

- Break #1 pc>\$e002 h
 - Coloca un breakpoint que se dispara y detiene la ejecución cuando se cumple que el PC es mayor que \$e002.

GO:

- Go \$e003 #1 :3
 - Inicia la simulación en modo continuo, desde PC=\$e003, se detiene en la 3° ocurrencia del breakpoint “1”.

Comandos del Simulador

UNTIL:

- Until p:\$e003 h
 - Ejecuta hasta la dirección \$e003 (inclusive) se detiene si hay algún breakpoint.

Reset:

- Reset d
 - Resetea el dispositivo (deja la memoria y los registros sin variación, inicializa el pc)

Ejercicios Manejo de Registros

Ejercicios 1

Inicializar los registros con los valores

a = \$ffffffffffff

b = \$ffffffffffff

x = \$ffffffffffff

Se ejecuta las siguientes instrucciones:

move #\$4b,x1

move #\$4b,a1

move #\$4b,b

Comentar el estado de los resultados

Ejercicios Manejo de Registros

Ejercicios 2

Inicializar los registros con los valores

a = \$ffffffffffff

b = \$ffffffffffff

x = \$ffffffffffff

Se ejecuta las siguientes instrucciones:

move #\$9b,x1

move #\$9b,a1

move #\$9b,b

Comentar el estado de los resultados

Comandos Compilador

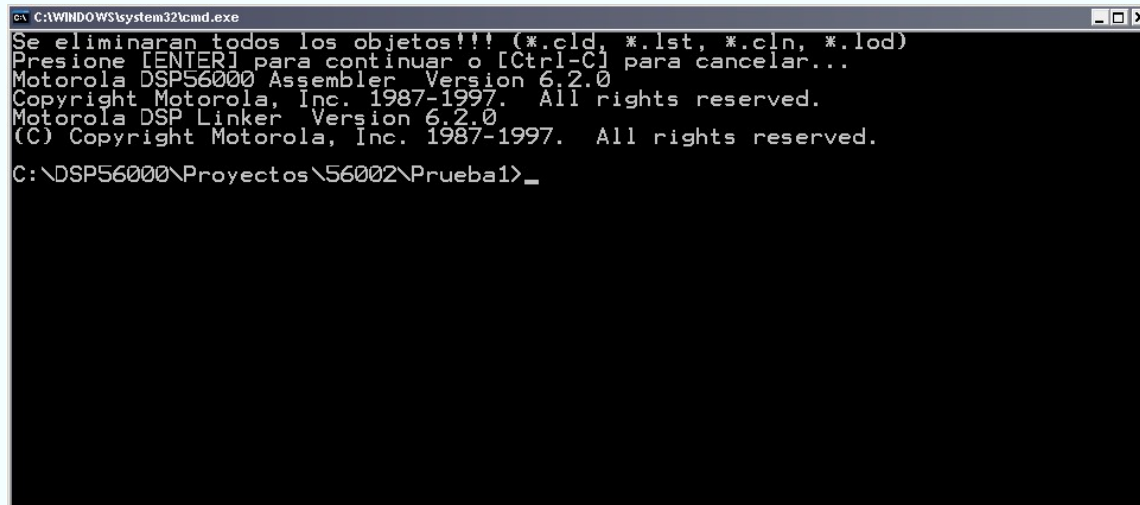


```
eje2.ASM - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda

LIST
ORG      P:$E000

main     move    #$caba00,x1
          move    x1,a
          move    x1,b1
          jmp     *
          end     main
```

C:...\>all eje2



```
C:\WINDOWS\system32\cmd.exe
Se eliminaran todos los objetos!!! (*.cld, *.lst, *.cln, *.lod)
Presione [ENTER] para continuar o [Ctrl-C] para cancelar...
Motorola DSP56000 Assembler Version 6.2.0
Copyright Motorola, Inc. 1987-1997. All rights reserved.
Motorola DSP Linker Version 6.2.0
(C) Copyright Motorola, Inc. 1987-1997. All rights reserved.
C:\DSP56000\Proyectos\56002\Prueba1>_
```

Ejercicios de Programación

Ejercicios 4

Analizar el Código del archivo eje4.asm

a.- ¿Cual es el objetivo del Código?

b.- ¿Que error tiene?¿Como se soluciona?

c.- Realizar la corrección del mismo y modificarlo para que la dimensión del programa se modifique atreves de la definición de "Var0"

Ejercicio de Programación

LIST	ORG	P:\$E000
Var0 EQU 2	main	clr a
org x:0		move #Var1,R0
Var1 equ *		move #Var2,R4
dc 1		move #VarRes,R5
dc 0		move x:(r0)+,x1 y:(r4)+,y1
dc 0		
dc 1		do #Var0,NewVar1
		rep #Var0-1
org y:0		mac x1,y1,a X:(r0)+,x1 y:(r4)+,y1
Var2 equ *	ActVar1	
dc 0.5		macr x1,y1,a X:(r0)+,x1 y:(r4)+,y1
dc 0.5		move a,y:(r5)+
		move #Var2,R4
VarRes equ *	NewVar1	
dc 0		
dc 0		jmp *
		end main