

Introducción al EMU8086

Facultad: Estudios Tecnológicos.

Escuela: Electrónica.

Asignatura: Microprocesadores

Objetivo General

Utilizar el programa emulador EMU8086 y relacionarlo con los elementos básicos que conforman el modelo de programación del microprocesador 8086.

Objetivos específicos

- Identificar los principales elementos que conforman el ambiente de programación del software EMU8086.
- Identificar los registros del modelo de programación del 8086.
- Observar la relación que existe entre los lenguajes de ensamblador y máquina.
- Reconocer de forma intuitiva los modos de direccionamiento básicos del micro 8086.
- Imprimir un carácter en la pantalla de usuario, modificando sus parámetros.
- Utilizar el emulador en la creación y ejecución un archivo tipo COM.

Materiales y equipos

- Guía de laboratorio
- Computadora PC con el programa EMU8086

Procedimiento

- 1) Ejecute el programa emulador de microprocesadores EMU8086, haciendo doble clic en el icono del escritorio, Aparecerá una ventana como la Figura 1.

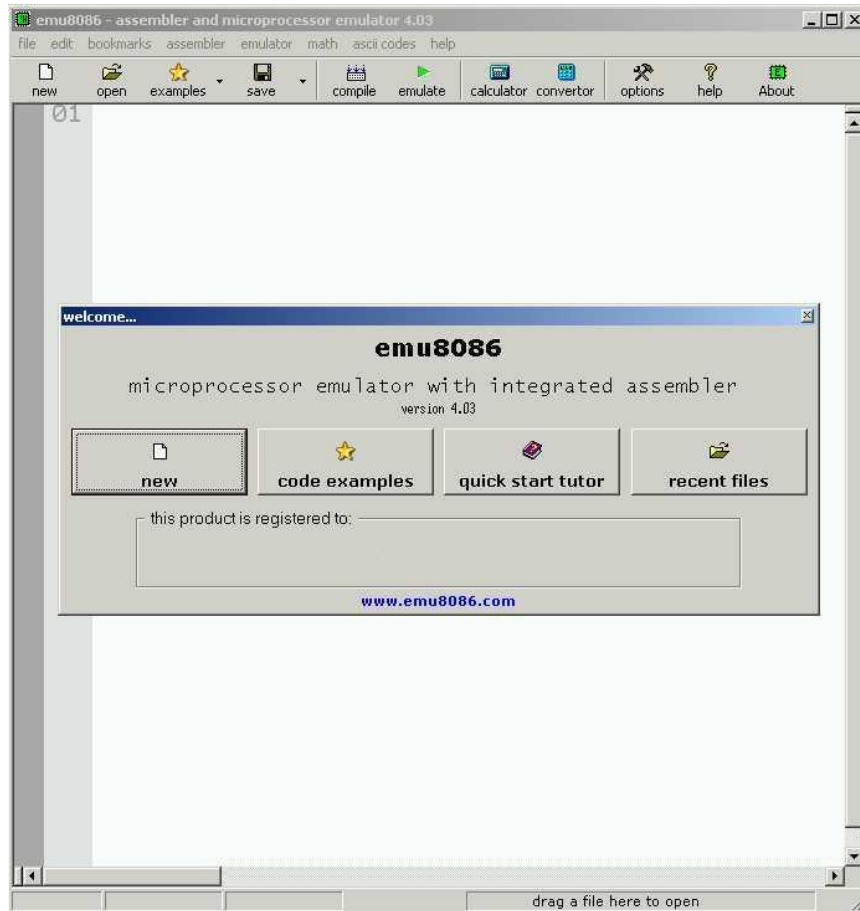


Figura 1: Ventana inicial del emu8086

- 2) Presione el botón que indica "New".
- 3) Observará una ventana donde escribirá sus futuros programas, llamados en general "código fuente".
- 4) Desde ahí se puede crear un nuevo documento. Observe la figura 2 "Choose code template"

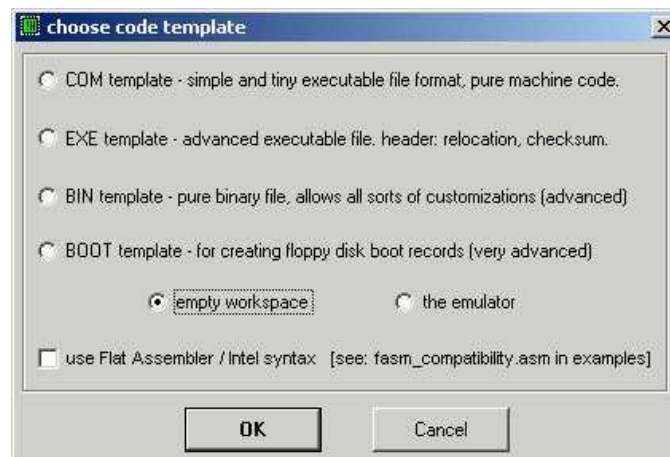


Figura 2

Se le presentarán seis opciones, elija "empty workspace". Observará (apóyese en la Figura 3) una ventana con la clásica barra de menú (File, Edit, etc.) con algunos elementos distintivos (Compile, Emulator, Math, etc.) así como una serie de botones que le permiten abrir un nuevo archivo, compilar, emular y otras funciones útiles. En el texto observará dos líneas de comando que son un encabezado necesario para crear un archivo tipo COM.



Figura 3

- 5) Seleccione en la barra de menú la opción "EMULATOR" y en ésta "SHOW EMULATOR". Se abrirá una ventana donde podrá observar diferentes elementos (apóyese en la Figura 4 para interpretar información). Algunos botones que permiten realizar acciones como: cargar (**load**), recargar (**reload**), ejecutar paso a paso (**single step**) y ejecutar (**run**). También podrá ver, en la parte central, tres columnas en las que se detalla el contenido de los registros del 8086, el contenido de la memoria y el programa desensamblado. Abajo podrá acceder a la pantalla de usuario (**user screen**), el código fuente (**actual source**), la Unidad Aritmética y Lógica (**ALU**), la Pila (**stack**) y el registro de banderas (**flags**) o estados.

- 6) Cierre la ventana del emulador, pero no cierre la ventana del código fuente.

NOTA: las direcciones de memoria están escritas en el formato de segmento y desplazamiento"

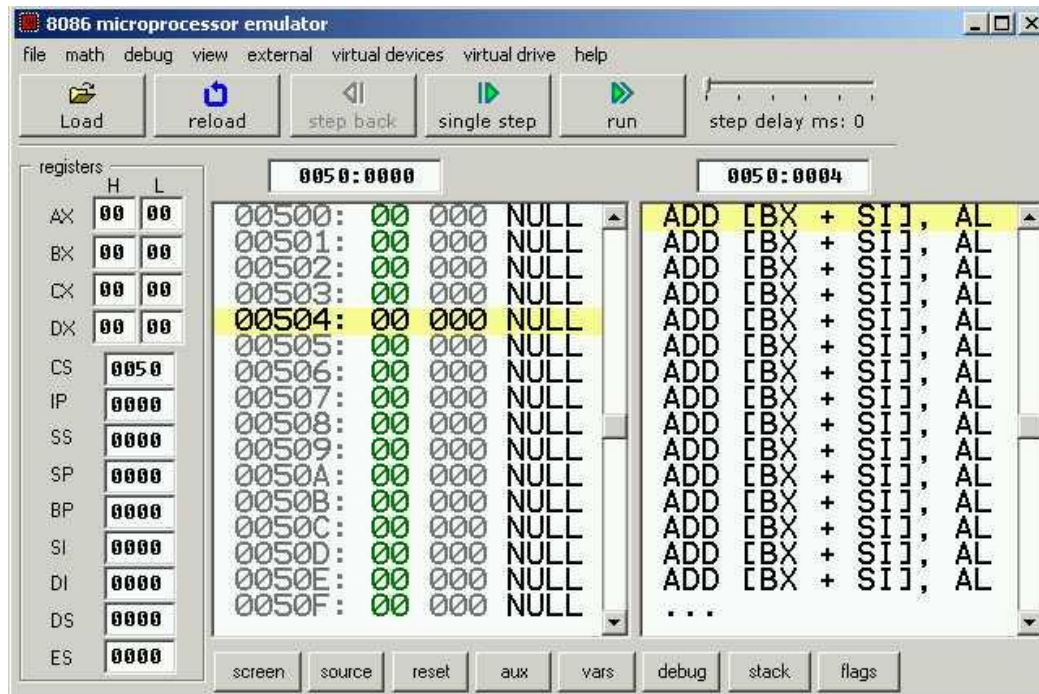
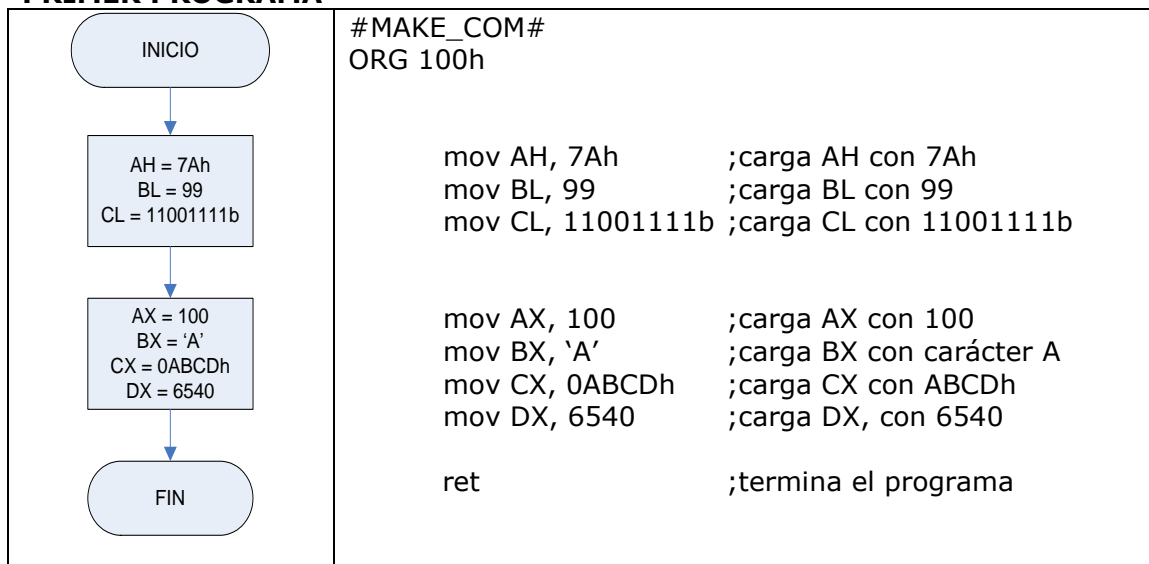


Figura 4

- 7) Ahora analice el siguiente código fuente (llamado PRIMER PROGRAMA), ayudándose de los comentarios y del diagrama de flujo que lo acompaña. Intente predecir cada cambio que tendrán los datos hexadecimales en los registros del microprocesador.

PRIMER PROGRAMA



- 8) Abra el archivo fuente llamado guia01.asm, como notará es el mismo programa que previamente analizó.
- 9) Usando el botón respectivo compile el código fuente, note que el programa le informa que creará un archivo con extensión COM.

¿Qué nombre le asigna el sistema al archivo .COM? _____

- 10) Si no se dan errores de compilación proceda a emular el programa. Analice la información que se le presenta auxiliándose de la Figura 5.
- 11) Puede observar una copia del código fuente en el que se marca la línea de instrucción que será ejecutada. También puede ver en la ventana del emulador el contenido de los registros, la memoria y el programa desensamblado. Si tiene problemas para interpretar esta información, por favor solicite asistencia por parte de su instructor.

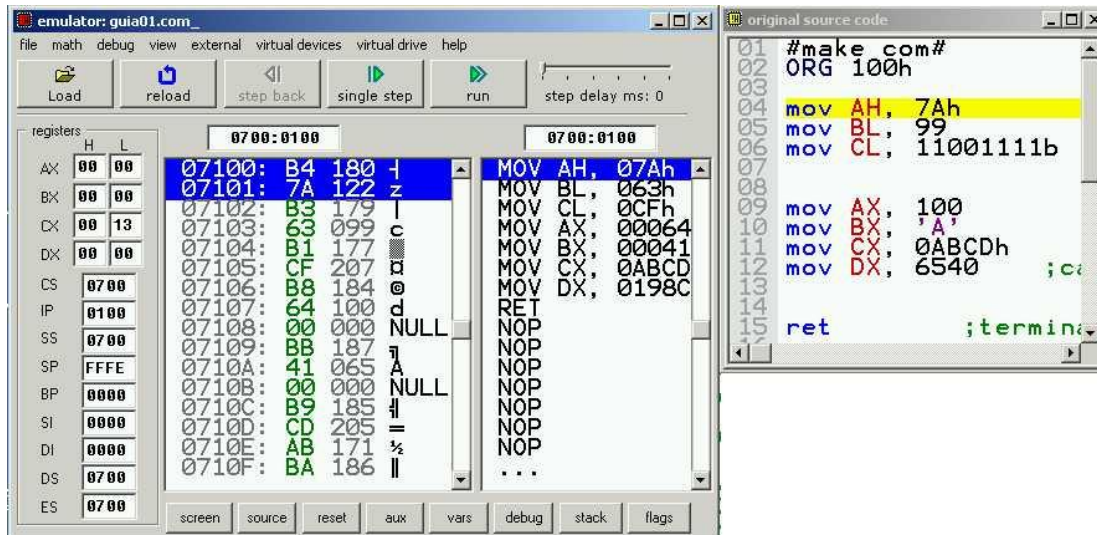


Figura 5

- 12) Copie los datos almacenados en los registros antes de ejecutar el programa:

AX:	BX:	CX:	DX:
CS:	DS:	ES:	SS:
BP:	IP:	SP	
DI:	SI:		

- 13) Ejecute el programa totalmente (RUN) y anote los datos resultantes. Verifique si concuerdan con los datos que usted esperaba.

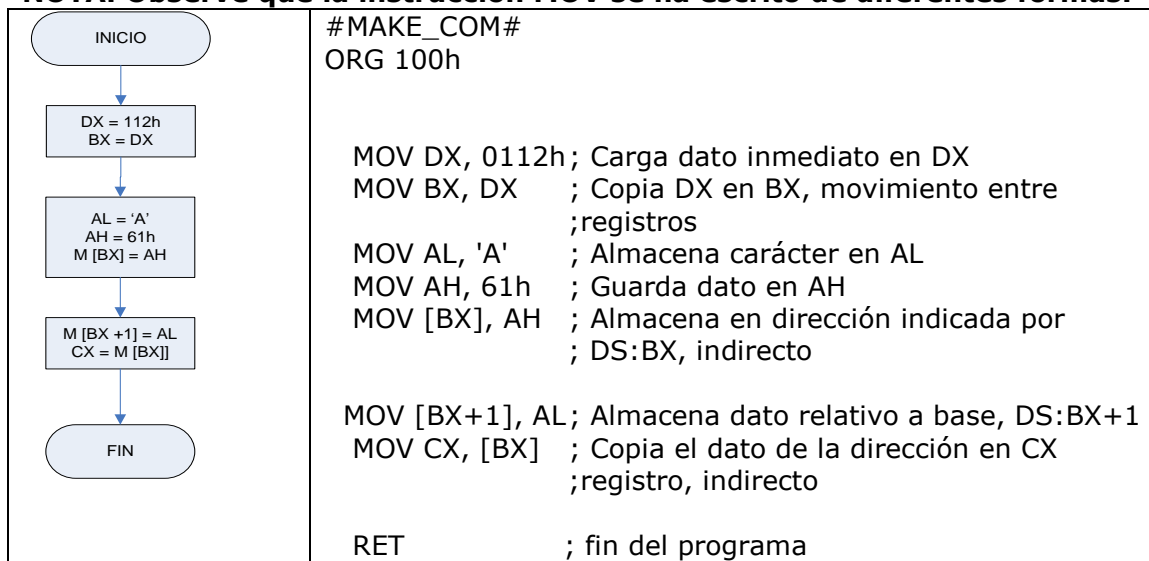
AX:	BX:	CX:	DX:
CS:	DS:	ES:	SS:
BP:	IP:	SP	
DI:	SI:		

- 14) Cargue nuevamente el programa, observe el contenido de los registros, sobre todo el contenido de CS e IP. Analice sus observaciones comparándola con la que aparece en la TABLA 1. Si tiene dudas por favor solicitar asistencia de su instructor.
- 15) Ejecute solo la primera instrucción (Single step). Observe los cambios (sobre todo en CS e IP) ¿se cargó el dato esperado en AH?
- 16) Interprete la información de la TABLA 1 respecto de sus resultados
- 17) Continúe ejecutando paso a paso el programa hasta la instrucción RET, al mismo tiempo llene los espacios vacíos de la TABLA 1. Si tiene dudas en esta actividad, por favor solicite la asistencia de su instructor.
- 18) Continúe ejecutando las instrucciones hasta concluir el programa.

DIRECCIONES		LENGUAJE DE MÁQUINA			LENGUAJE ENSAMBLADOR
SEGM(CS)	OFFSET	CAMPOS			LINEA
0B56	0100	B4	7 ^a		mov AH, 7A
					mov BL, 99
					mov CL, 11001111b
					mov AX, 100
					mov BX, 78
					mov CX, 0ABCDh
					mov DX, 6540
					ret
NOTA: A partir de este punto el programa ejecuta un procedimiento predeterminado llamado <u>Interrupción</u>.					
0B56	0000				INT 20H
F400	0150				

TABLA 1

- 19) Cierre la ventana del emulador, pero no cierre la ventana del código fuente.
- 20) Cree un archivo nuevo, tipo COM y escriba el código llamado SEGUNDO PROGRAMA, después de su respectivo encabezado, no olvide los tabuladores.

SEGUNDO PROGRAMA**NOTA: Observe que la instrucción MOV se ha escrito de diferentes formas.**

- 21) Emule el programa y antes de ejecutarlo anote en la TABLA 2 el contenido de las direcciones que tienen desplazamiento 0112h y 0113h, así como el contenido de CX en la columna Inicio. Más adelante se retomará este punto para comparar los cambios en los datos.
- 22) Corra paso a paso el programa hasta antes de ejecutar la quinta instrucción, tratando de predecir las acciones que se realizarán, auxiliándose de los comentarios de cada instrucción, si tiene problemas consulte con instructor.
- 23) Observe y anote en la TABLA 2 los resultados de su observación
- 24) Ejecute la quinta instrucción, anote los resultados.
- 25) Ejecute paso a paso el resto de instrucciones, tomando nota de los resultados.

SEG (DS)	OFFSET	Inicio del programa	Antes de 5ª	Después de 5ª	Después de 6ª	Después de 7ª
	0112h					
	0113h					
REGISTRO CX						

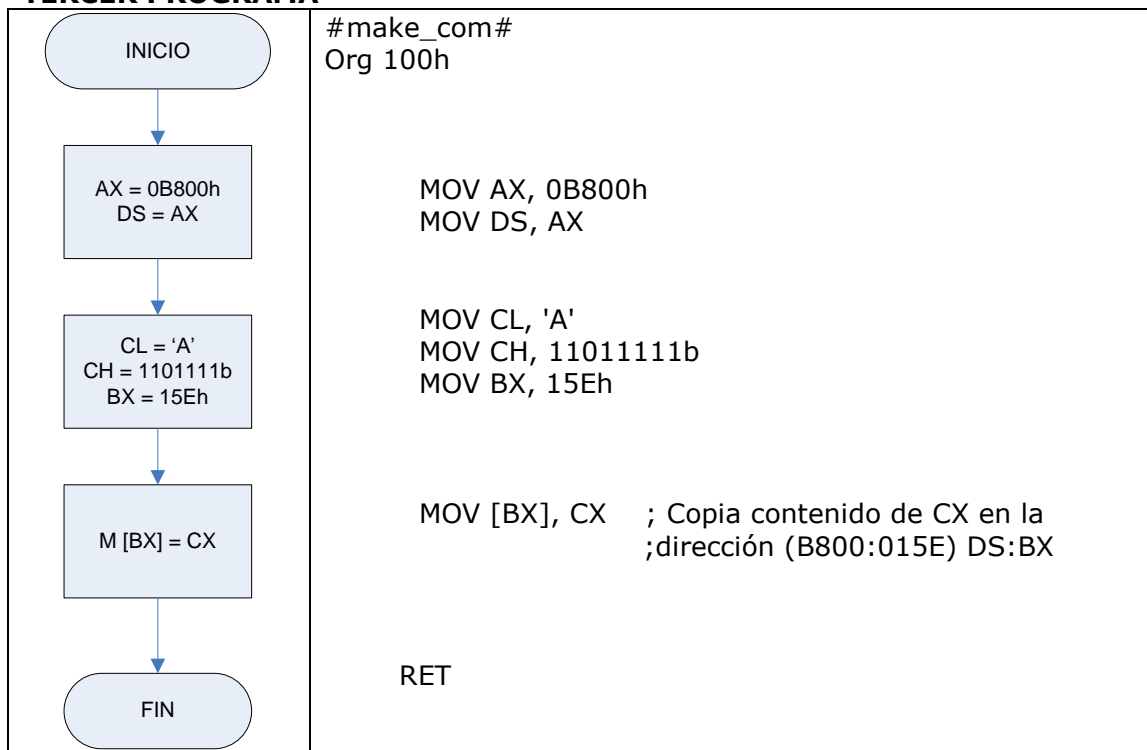
TABLA 2

- 26) Cargue nuevamente el programa.
- 27) Ejecute paso a paso el programa, anotando en la TABLA 3 el valor de cada dato que se le solicite, tomando como referencia el procedimiento con el que completó la TABLA 1.
- 28) Cierre la ventana del emulador, pero no cierre la ventana del código fuente.

DIRECCIONES		LENGUAJE DE MÁQUI			LENGUAJE ENSAMBLADOR
SEGM	OFFSET	CAMPOS			LINEA

TABLA 3

- 29) Escriba un nuevo programa .COM y digite el código llamado TERCER PROGRAMA.
- 30) Emule y ejecute el programa (run) y note la información que se presenta en la pantalla de usuario, debe observar un carácter 'A' con un color para el fondo y para el carácter. Anote esa información.

TERCER PROGRAMA

- 31) Modifique el programa sustituyendo el dato que se almacena en BX (15Eh) por 10Eh y corra el programa.
- 32) Observe y describa como afectó el cambio al funcionamiento del programa.

33) Modifique ahora el dato almacenado en CH, cambiándolo por 11010000b y observe su efecto, anote sus observaciones.

34) Vuelva a cambiar CH, ahora por 10011110b, observe y describa el cambio.

35) Pruebe a hacer algunos cambios en CH, CL y BX, anote sus observaciones:

36) Notifique a su instructor que ha terminado la práctica.

Análisis de resultados

a) Primer programa.

En el encabezado del programa ¿por qué se incluye la línea "org 100h"?

Al introducir datos en los registros del microprocesador se usan diferentes formatos (bases de numeración) ¿cuántos y cuáles son?

La sintaxis de la instrucción MOV define tres campos: instrucción, dato1 y dato2. Al moverse los datos ¿cuál es la fuente y cual el destino?

Al observar la tabla1 se nota que a la misma instrucción MOV se le asignan diferentes códigos hexadecimales ¿a qué se debe esto?

¿Cuál es la función que cumple el registro IP dentro del PRIMER PROGRAMA?

En la columna OFFSET de la tabla1, éstos no se incrementan a una razón constante ¿por qué?

b) Segundo programa.

Al comparar las dos primeras instrucciones ¿qué diferencias hay al ejecutarlas, siendo que las dos son MOV?

En las últimas tres instrucciones se incluyeron corchetes (paréntesis cuadrados) en las instrucciones ¿Qué diferencias observó en su ejecución respecto a las primeras dos instrucciones?

¿Por qué al ejecutar la séptima instrucción los datos se almacenan en CX en ese orden específico?

c) Tercer programa.

Al ejecutar el programa, a medida que hacía las modificaciones ¿qué efecto concreto logra el cambio en el registro BX?

Al modificar CH ¿qué bits específicos se modificaron para lograr cambios en el fondo y color de los caracteres?

Investigación complementaria

Esta actividad debe desarrollarse posteriormente a la ejecución de la guía de laboratorio y tiene por objetivo complementar y ampliar la temática desarrollada teniendo en mente las actividades futuras.

Investigue:

¿Qué son los modos de direccionamiento?

¿Cuántos y cuáles modos de direccionamiento tiene el microprocesador 8086?

¿Qué modos de direccionamiento posee la instrucción MOV?

¿Qué diferencias hay entre las instrucciones **XCHG** ya la instrucción **MOV**?

Bibliografía

Brey, B. B. [*Los Microprocesadores Intel. 8086 / 8088, 80186, 80286, 80386 y 80486. Arquitectura, programación e interfaces.*](#) Prentice Hall, México DF, 1995³
Biblioteca UDB 001.6404 B847 1997