

INTRODUCCION

AL

CPU08

Curso de Microcontroladores

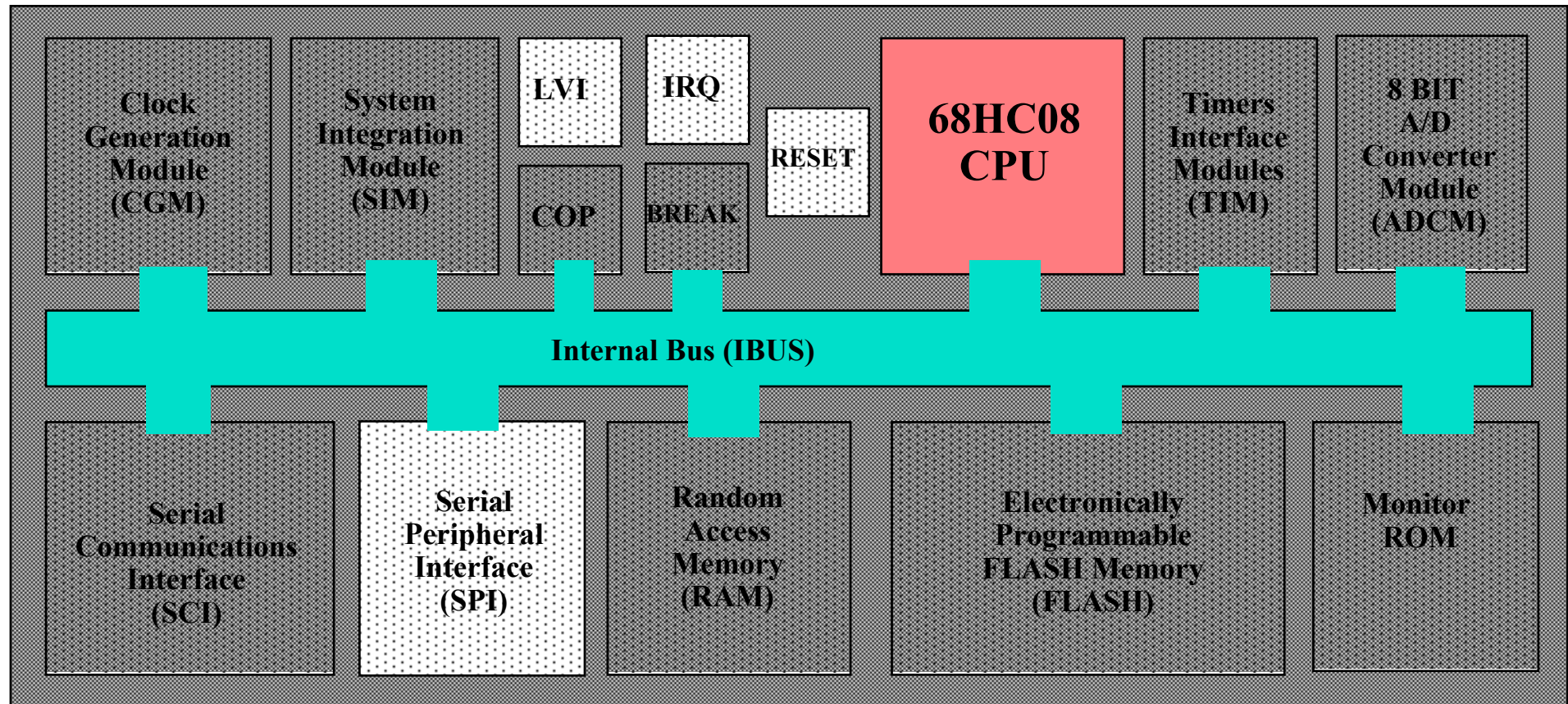
Familia HC908 Flash de Motorola

Parte II

CPU08 Core

ING. DANIEL DI LELLA DDFAE For Motorola Products

CPU08



EI CPU 08:

El CPU 08 es el “corazón” de los MCUs de la flia. HC908. Es sin dudas, una versión muy mejorada del CPU 05 corazón de la flia. HC705.

Dentro de la estructura interna de un HC908, el módulo del CPU se vincula con el resto de los módulos del MCU por medio de un BUS de DATOS interno de 8 bits, y un BUS de Direcciones de 16 bits, que le permiten direccionar código de hasta 64K bytes. Este Bus es denominado IBUS (Internal Bus).

La frecuencia máxima del BUS interno es de 8 MHZ reales a 5 Volts de alimentación y 4 MHZ a 3 Volts. Esta frecuencia de Bus implica que cada ciclo de Clock (reloj) del Bus es de 125 nS !!!, siendo esta una notable mejora con respecto a la familia HC705.

Recordemos que al ser el HC908 un MCU del tipo “cerrado” (single chip) NO se tiene acceso a los BUSES internos del mismo. No sucede lo mismo con la familia HC11 que puede trabajar tanto en modo “single chip” (cerrado, todo en el chip), como en modo “expanded” (expandido o abierto, osea disponer de los pines de Bus hacia el exterior) y de esta forma poder direccionar memorias externas.

Veamos ahora un poco la arquitectura de ejecución que presenta el CPU 08 :

CPU08

Arquitectura de ejecución

Curso de Microcontroladores

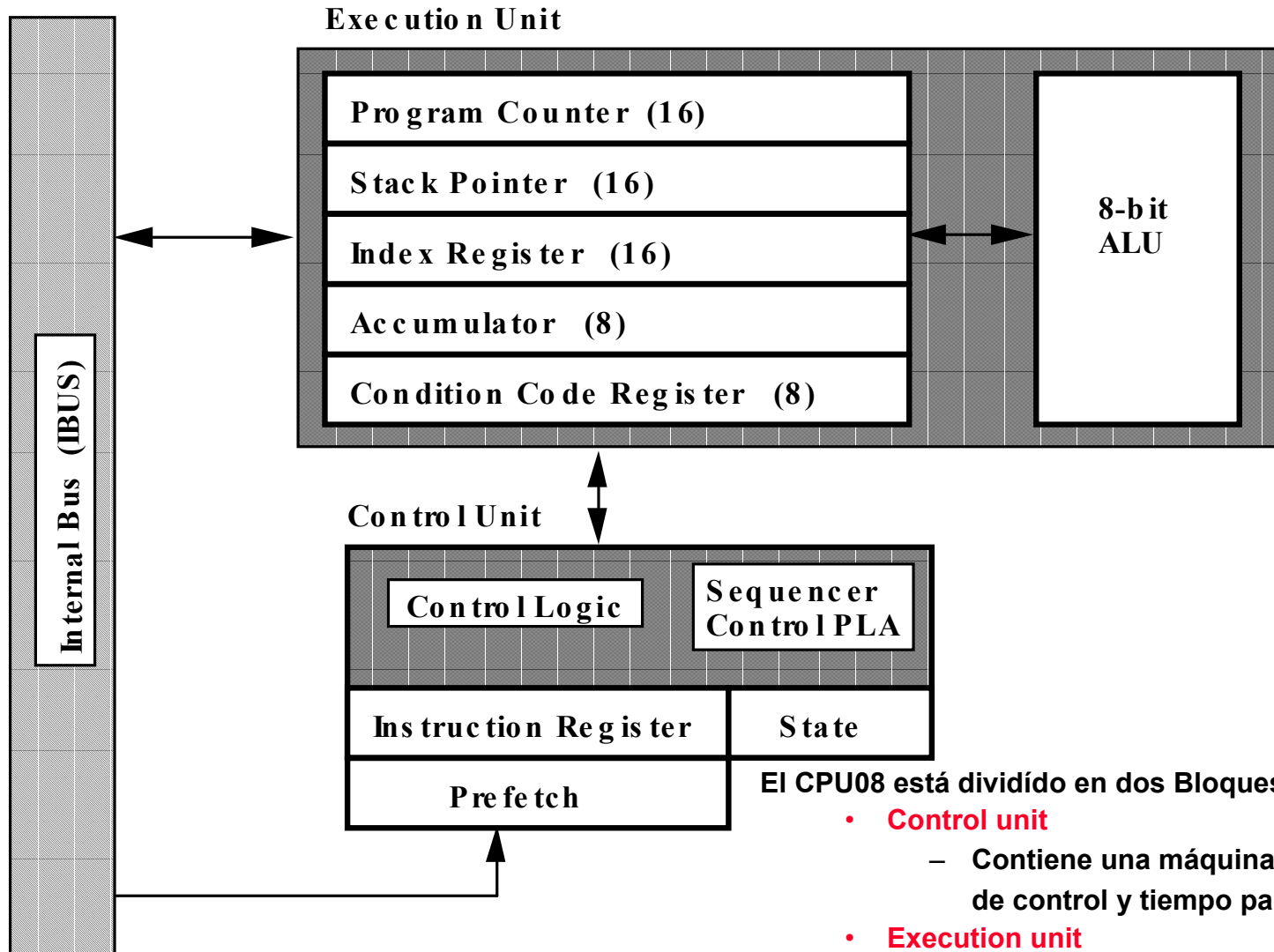
Familia HC908 Flash de Motorola

Parte II

CPU08 Core

ING. DANIEL DI LELLA DDFAE For Motorola Products

CPU08 Arquitectura de ejecución



El CPU08 está dividido en dos Bloques...

- **Control unit**
 - Contiene una máquina de estados finitos, unidades de control y tiempo para manejar la “execution unit”.
- **Execution unit**
 - Contiene la ALU, registros, el bus interface

Arquitectura de Ejecución:

El CPU 08 pertenece a la arquitectura del tipo “Von Neuman” clásica, característica de la familia 68xx de Motorola y ampliamente utilizada en el mundo.

En este tipo de arquitectura, existe un solo Bus de datos, tanto para memoria de programas como para memoria de datos, lo que da origen a una mapa “lineal” de acceso a memoria y por consiguiente **NO EXISTEN** instrucciones especiales y diferentes para trabajar con “DATOS” o con código de programa.

De esta forma, **TODAS** las instrucciones son aplicables en cualquier parte del mapa de memoria sin importar si se trabaja con datos o código. Por lo que no es raro encontrar aplicaciones cuyos programas corren desde RAM como si estuvieran en Flash. Esto no podría hacerlo una arquitectura Hardvar clásica.

El CPU está formado por dos grandes bloques, el “EXECUTION UNIT” y el “CONTROL UNIT”.

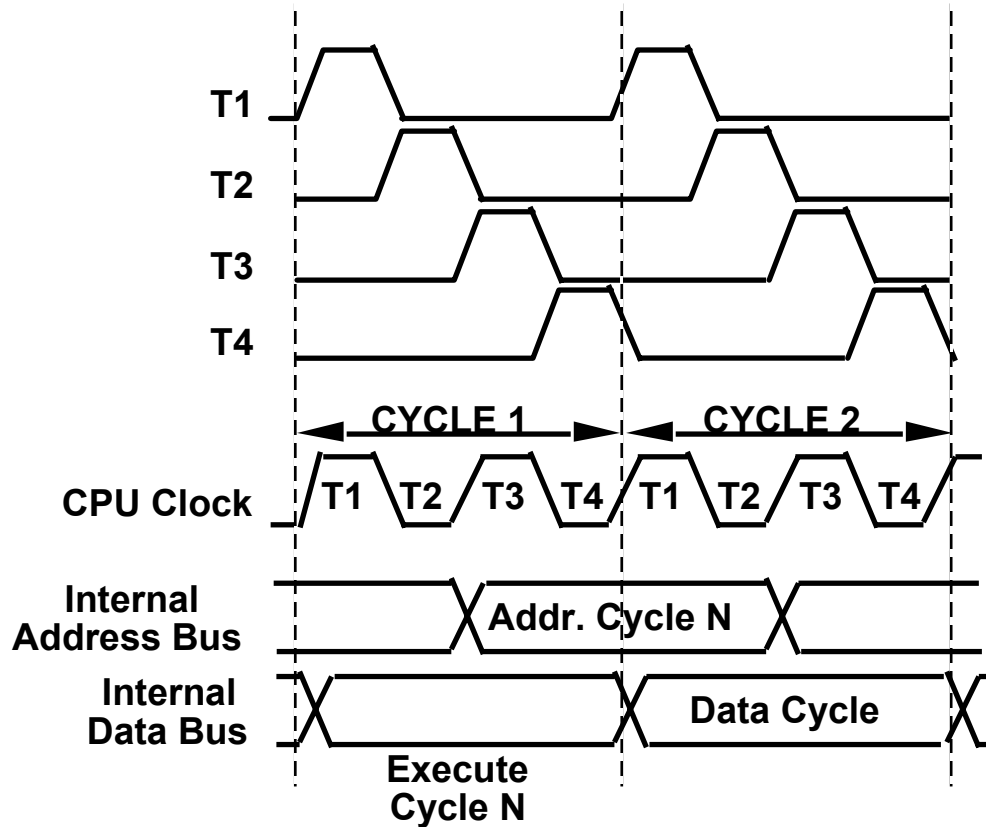
El Execution unit, contiene la Unidad Lógica y Aritmética (encargada de todas las operaciones lógicas binarias y aritméticas), Registros internos del CPU (Acumulador, Puntero de pila, Contador de Programa, Registro índice, Registro de Código de Condiciones) y la interface con el BUS interno.

El control unit, contiene una máquina de estados finitos , unidades de control y tiempo para “manejar” el execution unit.

Temporización Interna

La temporización del CPU08 se deriva desde un "clock" de 4 fases

El ciclo del CPU consiste de un pulso de clock desde cada fase



Cycle 1

T1, T2 - se calcula nuevo direc. P/ ciclo2

T2 - El Dato es leído p/ esa dirección.

El direccionamiento precede al dato 1/2 ciclo de Bus.

El “Prefetch” en el CPU08

El **CPU05** tiene muchos ciclos en donde el “address Bus” y el “Data bus” están **“IDLE” (sin actividad)**. Esto provoca que las distintas instrucciones del mismo consuman más ciclos de reloj de lo que sería normalmente necesario.

El **CPU08** contiene un “opcode” con un mecanismo de Pre-búsqueda hacia adelante.

- Se incrementa la Performance por remoción de tantos “ciclos muertos” de Bus como sea posible.
- De esta forma se obtienen instrucciones con menor número de ciclos de reloj, mejorando la velocidad real de ejecución de código.

El flujo de instrucciones del CPU08 fue desarrollado para ser tan eficiente como sea posible en una estructura del tipo “pipeline”.

Ejecución de Instrucciones

Todas las instrucciones:

- Se ejecutan en un número finito de ciclos de Bus.
 - Se sugiere ver cada instrucción para comparar el número de ciclos de Bus insumido por cada una de ellas.
- Carga el próximo opcode en el “Opcode Lookahead register”
- Incrementa el “Program Counter” (PC) a la próxima ubicación.
 - Sucede después del prefetch
 - El PC estará apuntando al byte siguiente del opcode “Pre-buscado”

Comentario general :

El clock del CPU08 es más complejo que el del CPU05, ya que necesita de un clock de 4 fases para ejecutar un ciclo de máquina, en lugar de los 2 necesarios en el CPU05. Por ello, en el HC908, se necesita un **XTAL** (Oscilador Externo) **4 veces superior a la frecuencia de Bus deseada** y el “doble” de lo usual en la familia HC705. Por ejemplo, para trabajar con una **Fbus de 8 Mhz**, se necesitará un cristal de **32 Mhz** para obtener la misma.

Gracias a una estructura interna del tipo “pipeline” en las distintas instrucciones del CPU08 se remueven tantos ciclos de máquina como sean posibles, ya que mediante un mecanismo de “pre-búsqueda” se minimizan los “tiempos muertos” (dead time) típicos de estructuras Von Neuman. El CPU05 NO fue concebido para minimizar los tiempos muertos de una instrucción, por lo que es frecuente en la familia HC705 ver al CPU en estados “IDLE” durante la ejecución de un programa.

Es por esta razón, que la mejora total de la performance en los HC908 es **5 (cinco) veces** superior a la de los HC705, a igual código ejecutado.

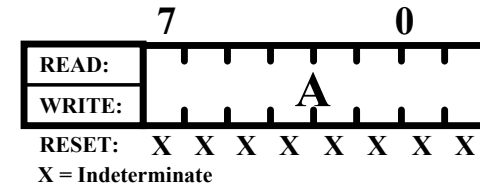
Señales de Control

Dos señales del “Control Unit” controlan el “prefetch” y la carga de instrucciones.

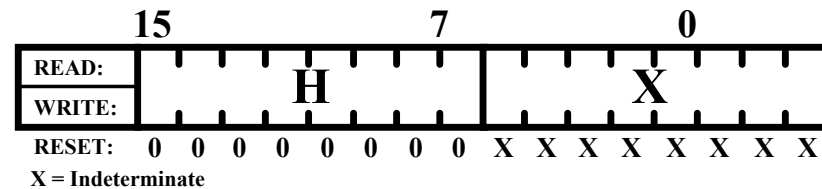
- **Opcode Lookahead**
 - Señales para la operación “prefetch”
- **Lastbox**
 - Señales para el último ciclo de la instrucción en curso

Registros

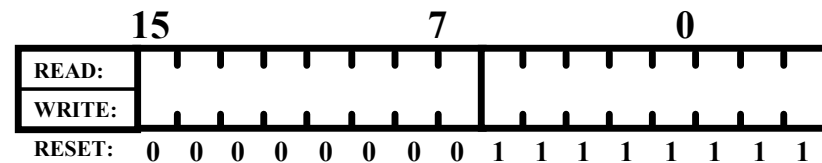
Acumulador (A)



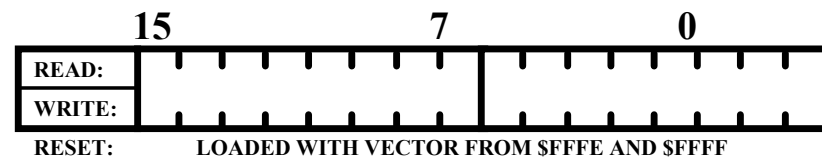
**Registro Índice (H:X)
(16 BITS)**



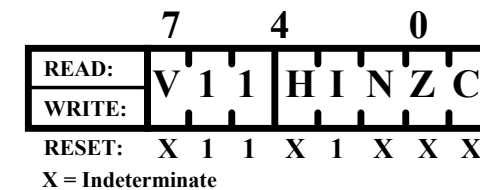
**Puntero de Pila (SP)
(16 BITS)**



**Contador de Programa (PC)
(16 BITS)**



Registro de condiciones de código (CCR)



Registros del CPU:

El CPU08 presenta los siguientes registros:

Acumulador (Accumulator, ACC) : Es el registro principal para operaciones matemáticas, de transferencia de información y operaciones lógicas. Es un registro de 8 bits de longitud sumamente usado en una estructura Von Neuman y por lo tanto existen muchas instrucciones que lo involucran.

Registro Indice (Index Register, H:X): Es el registro utilizado para todas las operaciones indexadas (con y sin offset) que posee el CPU. A diferencia del CPU05, en el CPU08 este registro es de 16 bits de longitud, ya que es la unión de los sub índices “X” y “H” (parte “baja” y parte “alta” del registro respectivamente). Esta característica hace posible el direccionamiento indexado de una longitud máxima de 64K bytes en lugar de los 256 Bytes máximos posibles en el CPU05. De esta forma se pueden armar tablas de hasta 64K bytes sin recurrir a “trucos” de paginado o similares.

Puntero de Pila (Stack Pointer, SP): Es el registro utilizado por el CPU para mantener en “orden” (guardar y rescatar los datos en RAM) los registros principales del mismo ante una excepción en la secuencia del programa, como lo son los saltos a sub-rutinas y los distintos pedidos de interrupción. Este tiene una longitud de 16 Bits, lo que permite No solo recorrer el espacio completo de memoria RAM, sino también espacio de memoria de programa (Flash). Esta característica permite el uso de este registro como “segundo registro indice” muy utilizado en los compiladores de lenguaje “C” para hacer más eficiente la generación de código compilado.

Contador de Programa (Program Counter, PC): Es el registro utilizado por el CPU para mantener el control de las direcciones de las próximas instrucciones a ser ejecutadas. Este registro tiene una longitud de 16 Bits, lo que permite un largo de programa teórico de 64K bytes. Dependiendo del derivativo utilizado, la memoria de programa podría ser inferior a los 64K, lo que en la práctica dá paso a lugares de memoria NO IMPLEMENTADOS (huecos de memoria) que si por alguna razón el PC apunta a esa dirección inexistente, genera un error de acceso detectado en forma automática por el CPU08 e indicado como “Illegal Adress” y la posterior generación de un “Reset” general del sistema como consecuencia de ello.

Registro de Condiciones de Códigos (Condition Code Register, CCR): Este registro es el utilizado por el CPU para “saber” como han resultado las operaciones aritméticas / matemáticas efectuadas por este con anterioridad. Los distintos Bits son utilizados como “Flags” (banderas) que indican el resultado de una operación determinada. Podemos observar que el CCR del CPU08 es mas completo que el de la flia. HC705, lo que permite tener mejor control de las distintas operaciones matemáticas y lógicas a efectuar.

BITs del C.C.R

	7		4		0			
READ:	V	1	1	H	I	N	Z	C
WRITE:								
RESET:	X	1	1	X	1	X	X	X

X = Indeterminate

V - Flag overflow del Complemento a dos

- “Seteado” si una operación aritmética signada ha desbordado
- Utilizado en “chequeos” de operaciones aritmética signadas

H - Half Carry flag

- “Seteado” si un carry (acarreo) ocurre desde el bit 3 al bit 4
- Utilizado en operaciones aritméticas BCD

I - Global Interrupt Mask (Mascara Global de Interrupciones)

- Cuando está seteada, deshabilita las interrupciones del CPU.

N - Negativo

- Seteado si el bit 7 está seteado en el Acumulador

Z - Zero flag

- Seteado si TODOS los bits en el Acumulador son ceros.

C - Carry or Borrow flag

- Seteado si un carry o borrow ocurren durante una operación

Fin Capítulo 3 !!