**APUNTES DE MICROPROCESADORES Y MICROCONTROLADORES**

Temario

1. Introducción a los microcontroladores (MCU) y Microprocesadores (MPU)
2. Estructura de un MPU
3. Estructura de un MCU
4. Elementos estructurales de la memoria
5. Programación del MPU
6. Interfase del MCU con dispositivos externos
7. Puertos de comunicación E/S paralelo
8. Puertos de comunicación serial asíncrona y síncrona
9. Temporizadores
10. Aplicaciones de los MPU y MCU

**BIBLIOGRAFIA**

|  |  |  |  |
| --- | --- | --- | --- |
|  | INTEL | **MOTOROLA** | **TEXAS INSTRUMENT** |
| NICHO | Computadoras personales | Control de procesos industriales | Procesamiento digital (DSP) |

**ARQUITECTURA VON NEWMAN**

Propuso dos conceptos básicos:

1. Utilización del sistema de numeración binaria
2. Almacenamiento de la secuencia de instrucciones de que consta el programa de una memoria interna, fácilmente accesible, junto con los datos de referencia, con esto se aumento la velocidad de proceso.

Un solo bus de datos y direcciones, el mismo bus se emplea para enviar y recibir instrucciones y datos. Los datos y las instrucciones son almacenados en una memoria principal, CPU 🡪va a la memoria principal, extrae las instrucciones y después los datos.

arquitectura harvard

Esta arquitectura se caracteriza por tener por separado el bus de datos y el bus de direcciones. Esto significa que las instrucciones y los datos son almacenados en memorias diferentes que son accedidas de forma separada por la CPU.

**MEMORIA PRINCIPAL**

**DATOS**

**+**

**INSTRUCCIONES**

**CPU**

**UNIDAD DE CONTROL**

**BUS DE DIRECCIONES**

**BUS DE DATOS E INSTRUCCIONES**

**BUS de CONTROL**

**MEMORIA PRINCIPAL**

**DATOS +**

**INSTRUCCIONES**

**CPU**

**DATOS +**

**INSTRUCCIONES**

**UNIDAD OPERATIVA**

**DATOS +**

**INSTRUCCIONES**

**ARQUITECTURA VON NEWMAN**

**MEMORIA PRINCIPAL**

**INSTRUCCIONES**

**CPU**

**UNIDAD DE CONTROL**

**MEMORIA DE INSTRUCCIONES**

**DATOS +**

**INSTRUCCIONES**

**CPU**

**DATOS +**

**INSTRUCCIONES**

**UNIDAD PERATIVA**

**DATOS +**

**INSTRUCCIONES**

**CPU**

**DATOS**

**MEMORIA DE DATOS**

**DATOS +**

**INSTRUCCIONES**

**ARQUITECTURA HARVARD**

elementos de una computadora

**UNIDAD DE MEMORIA**

**UNIDAD ARITMÉTICA LOGICA**

**UNIDAD DE CONTROL**

**UNIDAD DE ENTRADA**

**UNIDAD DE SALIDA**

MicroprocesadoRES

1. La unión de una Unidad Central de Control y la ALU, junto con algunos registros de transferencia forman la Unidad Central de Procesos o CPU de una computadora.
2. El concepto de CPU nace con las computadoras electrónicas de la primera generación fabricadas con tubo de vacío.
3. El concepto de Microprocesador nace con las computadoras electrónicas de la tercera generación fabricadas a base de circuitos integrados.

un microprocesador es un cpu integrado

ALU: procesa uno o dos números binarios efectuando operaciones aritméticas y operaciones lógicas.

unidad de control: responsable del control de todo el sistema, se encarga de proporcionar la secuencia y tiempo para el procesamiento de las instrucciones axial como la trayectoria y destino de los datos, se apoya en una señal de reloj el cual sincroniza el desarrollo temporal de todas las señales de control dentro y fuera del microprocesador. El sistema en cada instante en un estado perfectamente definido.

analogia: un policía de transito, dice quien pasa y quien no, su dirección y el momento en que deben hacerlo.

registro de control de condicion de estados (CCR): este registro proporciona información acerca de la última operación efectuada en la ALU, también conocido como registro de banderas.

acumuladores A y b: registros para almacenar operandos para ser procesados por la ALU.

registro de direccionamiento de memoria (MAR): registro para direccionar unívocamente localidades de memoria principal. Si 16 bits, 216= localidades direccionadas.

registro de direccionamiento de datos (MDR): registro para transferir datos desde o hacia la memoria principal.

registro de instrucciones (IR): registro para almacenar el código de la instrucción de cada operación efectuada por el microprocesador.

contador de programa (PC): registro para almacenar el número de la siguiente instrucción a ser procesada por el microprocesador.

registro x: registro para el manejo de índices, es como un directorio, con índice para acceder rápidamente y hacer tareas.

apuntador de pila (SP): registro para el manejo de subrutinas o interrupciones y otras operaciones.

comunicaciÒn entre modulos

La comunicación entre los módulos se efectúa en atención al tipo de información que se procesa. Puede ser de 3 tipos:

CONTROL

DATOS

DIRECCIONES

|  |
| --- |
| ¿ que es esto ?: se integraron en un solo chip y se le llama microcontrolador |
|  |
| μP+RAm+rom+puertos-E/S |

apunte anexo para la compreNsion de la mar y la mdr

**REGISTRO DE DIRECCION DE MEMORIA (MAR)**

**n lineas**

**DECODIFICADOR DE DIRECCIONES**

**(2n-1) líneas de direcciones**

Contiene la dirección que debe “abrirse “para los datos

Activa una sola línea de direcciones en la memoria

**REGISTRO DE DATOS DE MEMORIA (MDR)**

**(Salen los 8 bits)**

Esta diseñado de modo que se conecte adecuadamente todas las celdas en la unidad de memoria

PROBLEMA:

Para una memoria RAM de 2Kb y una ROM de 4Kb se usan como memoria principal de un microprocesador que tiene una longitud de palabra de 1 byte.

1. Determinar el número de palabras de la memoria RAM

1Kb=1024 bits

#bits totales RAM = 2(1024)=2048 bits --> 2048 [bits]/ 8 [bits/palabra]=256 [palabras]

1. Determinar el número de palabras de la memoria ROM

1Kb=1024 bits

#bits totales ROM= 4(1024)=4096 bits --> 4096 [bits]/ 8 [bits/palabra]=512 [palabras]

1. Numero total de palabras de la memoria total.

#palabras totales = =256+512 = 768 palabras

1. Número de líneas necesarias para el direccionamiento.

2#lineas=768

#lineas pertenece a los enteros.

#lineas = 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MICROCONTROLADOR Mcu | | | | |
| UNIDAD DE CONTROL | | | | |
| ES UN SISTEMA COMPLETO QUE INCLUYE EN UN SOLO chip: | | | | |
| para que sea MCU debe tener al menos estos elementos | MPU | | | |
| memoria (ROM,RAM,EPROM, E2PROM | | | |
| PUERTos de entrada-salida paralelo | | | |
|  | puertos de comunicación serie asincrona | | | |
|  | puertos de comunicación serie sincrona | | | |
|  | convertidor A/D | | | |
|  | UNIDAD DE TIEMPO (TIMERS) | | | |
|  | OSCILADOR Y RELOJ INTERNOS | | | |
| la memoria no es masiva ya que solo se requiere para control de procesos. | | | | |
| ESTRUCTURA DE UN MICROPROPROCESADOR | | | | |
| es el centro nervioso del mPU, proporciona la secuencia y tiempo para el procesamiento de las instrucciones, asi como para el control de la trayectoria y el destino de los datos. | | | | |
|  | | | | |
| ¿Cómo ES LA SECUENCIA? | En una instrucción en un **μP** se efectúa uno o mas ciclos de procesamiento (o ciclo máquina, o ciclo de procesamiento) | | | |
| CICLO DE PROCESAMIENTO (consta de dos etapas) | FETCH | Etapa de direccionamiento | Direccionamiento de una localidad de memoria principal vía **MAR** y transferencia de datos entre el **MDR** y la memoria principal | |
| EXECUTE | Etapa de ejecución | Se efectúa la operación indicada. | |
| El formato para especificar una instrucción esta compuesto por dios campos fundamentalmente | **Campo** para especificar la **instrucción** | | OPERADOR | Al menos un ciclo |
| **Campo** para especificar los **datos** | | OPERANDO | Uno o más ciclos |
| Cada operador y operando ocupan un lugar especifico en la memoria principal de forma tal que c/u de ellos, tiene que ser direccionado en forma secuencial y uno a la vez | | | | |

CICLO DE PROCESAMIENTO Y CICLO DE RELOJ DEL MPU

Sea F=2[MHz]=(1/4)\*8[MHz]

T=0.5[μs]

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | | | |
| 0.5 [μs] | | 0.5 [μs] | |
| **FETCH** | | **EXECUTE** | |
| 1 [μs] | | | |
| CICLO DE PROCESAMIENTO | | | |

El ciclo de procesamiento se ejecuta en 1[μs]

No confundir el ciclo de procesamiento con el ciclo de reloj del mpu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EJEMPLO: | | | | | | | |
| De acuerdo a la siguiente instrucción del **μP** especificado determinar el número de ciclos necesarios para que se efectúe la instrucción y lo que sucede en cada ciclo. | | | | | | | |
| mPU 🡪 8 bits  memoria principal 🡪 65kBYTE | | | | | | | |
| instrucción: suma contenido de la localidad de memoria $04b5 al contenido del acumulador y coloca el resultado en el acumulador | | | | | | | |
| FALTA DIAGRAMA ILUSTRATIVO DE LAS OPERACIONES | | | | | | | |
| 4 ciclos de trabajo | | | | | | | |
| FETCH | EJECUTE | FETCH | EJECUTE | FETCH | EJECUTE | FETCH | EJECUTE |
| Pc direcciona localidad 65 via MAR pc 🡪 0065 y su contenido se va al registro de instrucciones via MDR | Se ignora | Se direcciona 0066 por pc via MAR , PC se incrementa y su contenido se va al resgistro de instrucciones via MDR | Se ignora | Pc direcciona el siguiente via MAR, su contenido se va al registro de instrucciones via MDR | Se igniora | Se direcciona pc con 04B5 por pc, via MAR y como el registro de intrucciones esta completo se transfierea a la ALU via MDR | Hace la suma la ALU. |
| ciclo de PROCESAMIENTO 1 | | ciclo de PROCESAMIENTO 2 | | ciclo de PROCESAMIENTO 3 | | ciclo de PROCESAMIENTO 4 | |
| t1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |

Señal: cualquier magnitud física de naturaleza eléctrica codificada en código binario.

Este código puede ejecutarse directamente en el microprocesador.

Lenguaje maquina:

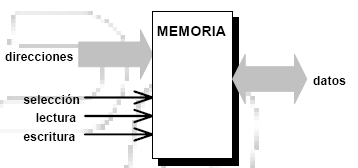
Una reducción significativa en la dificultad de programación se consiguió con la ejecución de los lenguajes ensambladores.

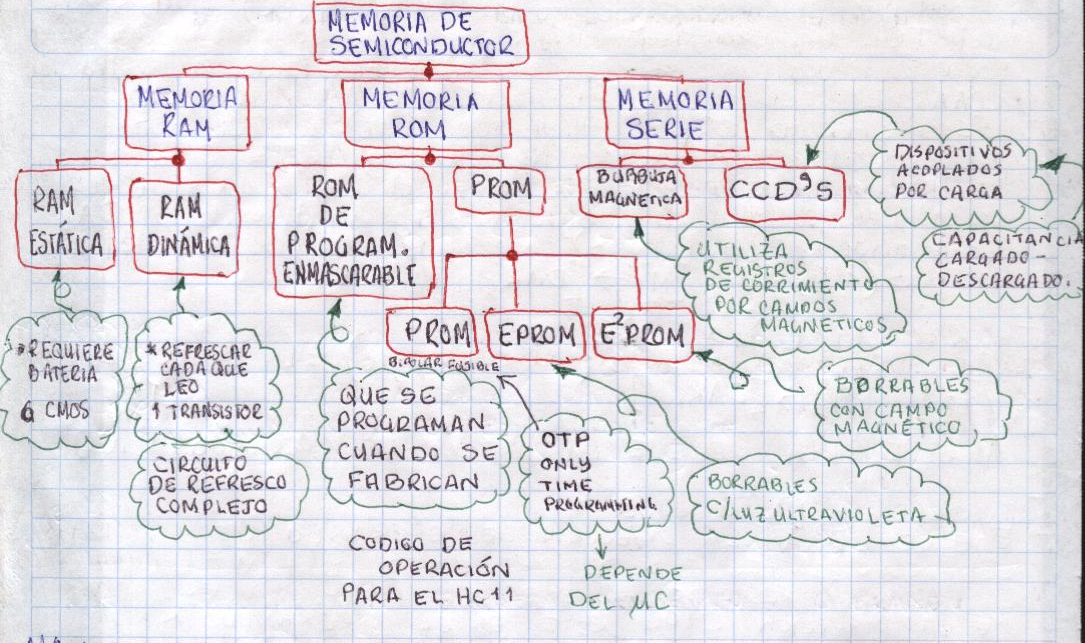
Lenguaje ensamblador: el lenguaje que permite escribir instrucciones en forma simbólica utilizando nombres fácilmente recordables (nemotécnicos).

ESTRUCTURA DE LOS ELEMENTOS DE MEMORIA

ESTADO ACTUAL DE LA TECNOLOGÍA

Esquema básico de una memoria





MEMORIA DE SEMICONDUCTOR

MEMORIA RAM MEMORIA ROM MEMORIA SERIE

MEMORIA DE SEMICONDUCTOR

MEMORIA RAM

MEMORIA ROM

MEMORIA SERIE

RAM ESTATICA

RAM DINAMICA

ROM DE PROGRAMACION ENMASCARABLE

PROM

PROM

EPROM

E2PROM

BURBUJA MAGNETICA

CCD’S

**PROGRAMA EN LENGUAJE ENSAMBLADOR**

**ENSAMBLADOR (traductor)**

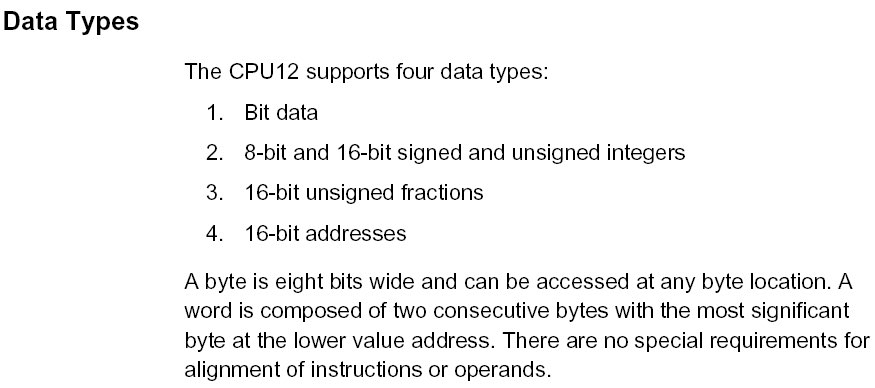
**LENGUAJE MÀQUINA**

**(programa objeto)**

**PROGRAMA EN LENGUAJE DE ALTO NIVEL**

**COMPILADOR**

**LENGUAJE DE MAQUINA**



MODOS EN QUE TRABAJA EL MICROCONTROLADOR HC11

|  |  |  |  |
| --- | --- | --- | --- |
| MODO | MODO B | MODO A | ¿QUE HACE? |
| single chip | 1 | 0 | El mC funciona como tal, es decir, con toda la disponibilidad de sus puertos y con un alcance para el usuario definido por su memoria interna |
| EXPANDED MODE | 1 | 1 | El mC funciona como microprocesador, direccionando una memoria externa, la cual s direcciona vía el bus de expansión de direcciones, perdiéndose por lo tanto los puertos D y C del mP. Así, quedan disponibles 16 pines para direccionar una memoria exterior hasta 64Kb |
| BOOTSTRAP | 0 | 0 | Variación del modo single chip, modo que ejecuta un programa bootloader en una memoria ROM interna llamada bootstrap. La memoria contiene el programa bootloader y conjunto especial de vectores de interrupción y reset. |
| TEST | 0 | 1 | Modo especial de prueba para los recursos del mC, ejecutado por el fabricante. |

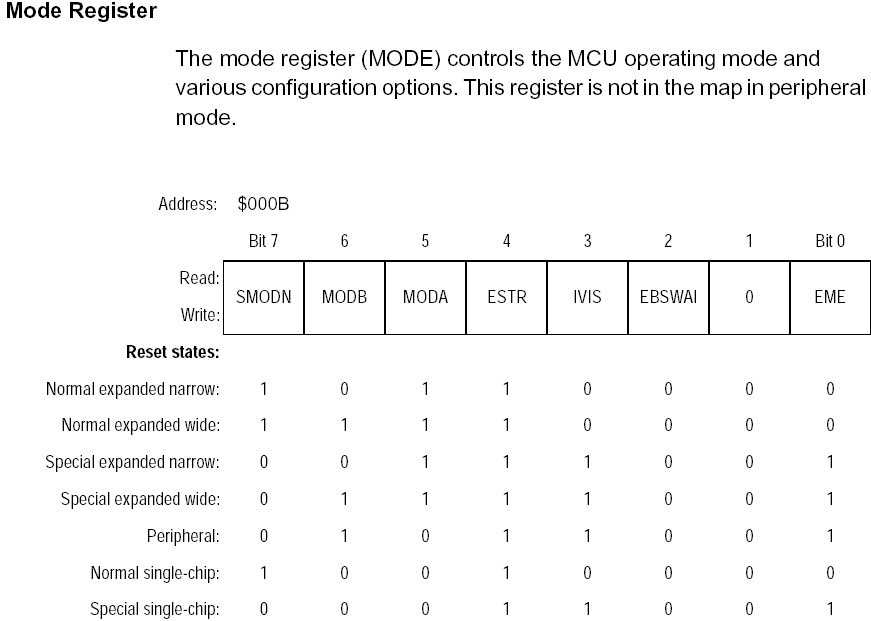
MODOS EN QUE TRABAJA EL MICROCONTROLADOR HC12

programacion del microcontrolador y el microprocesador

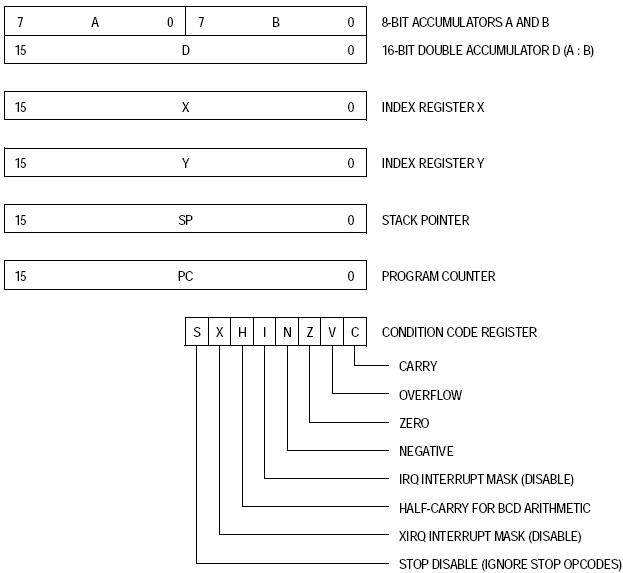
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| formato de instrucciones | | | | | |
|  | operador | tipo de  direccionamiento | base numerica | operando | OPERANDO |
| Mnemónico | Alfabético | ASCCI | ASCCI | \*según base numérica | \*según base numérica |
| Código de  Operación | Hexadecimal | ----------------------- | -------- | Hexadecimal | Hexadecimal |
| Numero de  Bits | 8 | --------- | ------ | 16 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TAMAÑO DE LAS INSTRUCCIONES | | | | |
| tamaño de las instrucciones | tipo de  direccionamiento | | notacion | |
| 1 byte (SIN OPERANDOS) | inherente |  | ------ | |
| 2 byte (con operandos) (1+1) | **DIRECTO** | dato reside en memoria | ------ | |
| inmediato | Dato esta junto a operador | # | |
| indexado | Numero que se ha de sumar al registro índice junto al operador | **\*\*,X (\*\*** de referencia) | |
| relativo | Estas indican a la CPU  que realice un salto de bytes hacia adelante o hacia atrás. El desplazamiento tiene signo y es de un byte por lo  que las bifurcaciones sólo se pueden hacer de 128 bytes hacia atrás ó 127 bytes hacia delante. | **\*,±H** (**\*** desplazamiento) | |
| 3 BYTE (con dos operandos) (1+2) | extendido | El dato se encuentra en la dirección de memoria especificada. El dato puede estar en cualquier posición de  la memoria dentro del límite de las 64Kb, por lo que la dirección ocupa 2 bytes. | -- | Condicionado |
| Incondicionado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| base numerica | | | | |
| decimal | | --- | | |
| hexadecimal | | $ | | |
| octal | | @ | | |
| binario | | % | | |
|  | | | | |
| LDAB |  | # | $ | 20 |
|  |  | Tipo de direccionamiento  (inmediato) | Base numérica |  |
| LDAB |  |  | $ | 20 |
| LDAB |  |  | $ | B3 |



MODELO DE PROGRAMACION DEL HC12



cosas interesantes acerca de un microprocesador.

una forma tipica de medir el rendimiento es por el tiempo de ejecucion de los programas:

rendimiento = 1/(tiempo de ejecución)

mips: millones de instrucciones por segundo

mflops: millones de operaciones en coma flotante por segundo

existen una serie de test que permiten medir el rendimiento de todo el sistema en conjunto.