Jesus Alberto Martinez Jimenez

A01227099-25/Agosto/2013

Organización computacional

Intel 8086

The instruction set of Intel 8086 is the ‘x86’, with the pass of the years the capacity of Intel 8086 have increased, the 16 bits was introduced in 1978, then in 1985 was introduced the 32 bits version and in 2003 almost after 20 years finally the 64 version could be possible, and with this last version the memory address registers capable more than 4GB using this new x86-64bits version. There are 6 instructions in the instruction set. The instruction format range is between 1 to 17 bytes, mostly due to all the complex adressing modes supporting and this mean more work for hardware and the assembler, instruction decoding is complex and its harder to compute the address of an arbitrary instruction.

Example:

; Sample 8086 assembly language program. This program

; prints the printable characters in a null-terminated

; string (similar to the unix ("strings" program).

; There is only one "segment" called "code" and the

; linker can assume DS and CS will be set to the right

; values for "code". The code begins at offset 100h

; within the segment "code" (MS-DOS .COM files).

code segment public

assume cs:code,ds:code

org 100h

start:

mov bx,offset msg ; bx points to string

loop:

mov al,[bx] ; load a character into al

cmp al,0 ; see if it’s a zero

jz done ; quit if so

cmp al,32 ; see if it’s printable

jl noprt ; don’t print if not

call printc ; otherwise print it

noprt:

inc bx ; point to next character

jmp loop ; and loop back

done:

int 20h ; return to DOS

; subroutine to print the byte in al

printc:

push ax ; push ax and dx

push dx

mov dl,al ; use DOS to

mov ah,02H ; print character

int 21H

pop dx ; restore ax and dx

pop ax

ret

msg db ’This’,9,31,32,’is’,20H,’a string.’,0

; example of how to reserve memory (not used above):

buf db 128 dup (?) ; 128 uninitialized bytes

code ends

end start

Freescale 68HC12

The instruction set of Freescale 68HC12 have these features it has a super set of the M68HC11 instruction set, also the implemetations of the original M68HC11 and the newer HC12, the memory and I/O are assign in a common 64-kbyte address space, this allow the same set of instructions to be used to access to the memory or I/O and control all the registers, and it has a full set of 8-bit/16-bits mathematical instructions(arithmetic, multiplication and division).The operands results are located in the memory. The 68HC12 instruction set consists of one byte (sometimes two bytes) of opcode and zero to five bytes of operand addressing information.

; 1+ ( n -- n+1 )

ONEP

LDD 0,X

ADDD #1

STD 0,X

RTS

; 2+ ( n -- n+2 )

TWOP

LDD 0,X

ADDD #2

STD 0,X

RTS

; 1- ( n -- n-1 )

ONEP

LDD 0,X

SUBD #1

STD 0,X

RTS

; 2- ( n -- n-2 )

TWOP

LDD 0,X

SUBD #2

STD 0,X

RTS

Microchip pic16f877a

The Microchip pic16f877a is orthogonal and is composed on three categories, byte oriented operations, bit oriented operations and literal and control operations, each instructions is a 14bit word divided in an opcode that specifies an instruction type and one or more operands that do a specify operation of the instruction, byte oriented instruction f is a file for register designator and d is a destination and this specify which one is going to use by the instruction. Bit oriented instruction b is a bit field designator that selects the bit affected by the operation while f is the address of the file where the bit is located. Literal and control operation k is an eight or eleven bit constant or a literal value.

#include<pic.h> // Blinking LED

#include<htc.h>

\_\_CONFIG(HS & PWRTEN & BOREN & LVPEN & WDTDIS & DUNPROT & DEBUGDIS);

void main(void)

{

unsigned int i;

ADCON1 =0x06 ; // Changes PORTA to digital

CMCON = 0x07 ; // Disable analog comparators

TRISA = 0x00 ; // Configure PORTA as output

PORTA = 0x00;

TRISD = 0X00 ;

PORTD = 0X00 ;

while(1)

{

//PORTA = 0XFF ;

//PORTB = 0XFF ;

//PORTC = 0XFF ;

PORTD = 0XFF ;

for(i=0;i<1500;i++);

PORTD = 0X00 ;

for(i=0;i<1500;i++);

//for(i=0;i<500;i++);

//for(i=0;i<500;i++);

}

}

Referencias:

<http://www.freescale.com/files/training_pdf/22422_HCS12_INSTRUCT_SET_WBT.pdf?lang_cd=en>

<http://en.wikipedia.org/wiki/X86-16>

<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010242>

<http://mai.kvk.uni-obuda.hu/documents/tantargy/8086-instruction-set-overview.pdf>

<http://www.edaboard.com/thread62300.html>

<http://www.mlsite.net/8086/>