

บันทึกกันลืม

เทคนิคการเขียนโปรแกรม Assembly

ATMEL AVR เบอร์ AT90S8535/2313/xxxx series

1. อย่าลืมคำสั่ง include file หรือใส่ชื่อไฟล์ไม่ตรงกับเบอร์ชิพียู ถึงคอมไพล์ผ่านแต่จะรันไม่ออก

.include "2313def.inc" ; สำหรับ AT90S2313

.include "8535def.inc" ; สำหรับ AT90S8535

1a. การกำหนด variable และ constant มีรูปแบบตามตัวอย่าง ดังนี้

.def temp = r16 ; กำหนดตัวแปรใช้เรียกแทน register

.equ speed0 = 0x00ff ; กำหนดตัวแปรใช้เรียกแทน ค่าคงที่ แบบ hexadecimal

.equ speed1 = 200 ; กำหนดตัวแปรใช้เรียกแทน ค่าคงที่ แบบ decimal

.equ segment1 = 0b01010111 ; กำหนดตัวแปรใช้เรียกแทน ค่าคงที่ แบบ binary

.equ text1 = 'A' ; กำหนดตัวแปรใช้เรียกแทน ค่าคงที่ แบบ ASCII

.equ relay_port = portb ; กำหนดตัวแปรใช้เรียกแทน ตัวแปรจากไฟล์ XXXXdef.inc

.equ relay1 = 1 ; กำหนดตัวแปรใช้เรียกแทน bit1 ()

1b. การกำหนดค่า bit address ให้ใช้เครื่องหมาย ',' คั่นเหมือนเป็น operand ที่สอง ตามตัวอย่างคำสั่ง ดังนี้

```
cbi    portb,1    ; clear bit i/o memory ของ portb ที่ bit1
```

2. register และ internal SRAM

- r16- r31 ให้ใช้เป็น accumulator ทุกตัว ใช้คำนวณคณิตศาสตร์ต่างๆได้เหมือนกันหมด

- r0- r15 ให้ใช้เป็น register ปกติ ใช้เก็บ/ฝากข้อมูลชั่วคราว

และอาจใช้คำนวณเหมือน r16-r31 ได้เมื่อมี operand เป็น register ด้วยกัน

- X,Y,Z = r27:r26, r29:r28, r31:r30 ตามลำดับ ใช้เป็น index register ขนาด 16bit

- internal SRAM แบ่งเป็น 3 กลุ่ม คือ

address: \$0000 - \$001F = r0 - r31 (working register)

address: \$0020 - \$005F = \$00 - \$3F (input/output memory)

address: \$0060 - \$015F/\$025F = \$0060 - \$015F/\$025F (internal ram ปกติ)

3. i/o memory ได้แก่ register ที่ทำหน้าที่เฉพาะทั้งหมด อันได้แก่

- SREG, SPH, SPL

- EEPROM register

- DDRx register, PORTx register, PINx register

- UART register, SPI register, ADC register, timer/counter register, watchdog register

- การย้ายข้อมูลเข้าหรือออกจาก i/o memory ต้องใช้ คำสั่ง out p,rr และ in rr,p เท่านั้น
ไม่สามารถใช้คำสั่ง load, store, move ได้

4 Data transfer ระหว่าง register, internal memory และ program memory ให้ใช้คำสั่งดังนี้

- mov rr, rr ; move register, register rr <-- rr
- in rr,p ; input register, port หรือ i/o memory rr <-- p
- out p,rr ; output port หรือ i/o memory, register p <-- rr

- ldi rr, k ; load register, constant rr <-- 0xkk
- lpm ; load r0, program memory ผ่าน Z r0 <-- (Z program memory)

- lds rr,\$skkkk ; load register, internal SRAM rr <-- \$skkkk
- sts \$skkkk,rr ; store internal SRAM, register \$skkkk <-- rr

- ld rr,X หรือ ld rr,X+ หรือ ld rr,-X ; load register, internal SRAM ผ่าน X
- ld rr,Y หรือ ld rr,Y+ หรือ ld rr,-Y หรือ ld rr,Y+q ; load register, internal SRAM ผ่าน Y
- ld rr,Z หรือ ld rr,Z+ หรือ ld rr,-Z หรือ ld rr,Z+q ; load register, internal SRAM ผ่าน Z

- st X,rr หรือ st rr,X+ หรือ st rr,-X ; store register, internal SRAM ผ่าน X

- st Y,rr หรือ st rr,Y+ หรือ st rr,-Z หรือ std rr,Y+q ; store register, internal SRAM ผ่าน Y
- st Z,rr หรือ st rr,Y+ หรือ st rr,-Z หรือ std rr,Z+q ; store register, internal SRAM ผ่าน Z

5. มี 9 คำสั่งที่ใช้ r16-r31 เป็นตัวเก็บผลลัพธ์ได้เท่านั้น ใช้ r0-r15 ไม่ได้

คำสั่งเหล่านี้จะมี operand รองเป็นค่าคงที่ทั้งหมด ซึ่งได้แก่

- ldi rr, kk
 - andi rr, kk
 - ori rr, kk
 - subi rr, kk
 - sbci rr, kk
 - cpi rr, kk
 - cbr rr, kk ; clear bit register, kk
 - sbr rr, kk ; set bit register, kk
 - ser rr ; rr <-- 0xff
- [แต่ clr rr ; rr <-- 0x00 rr = r0-r31 ได้ !!!!!]

6. คำสั่งคำนวณพิเศษมี 3 คำสั่ง คือ

- adiw rdl, k ; rdh:rdl <-- rdh:rdl + k เมื่อ k = 0.... 63
- sbiw rdl, k และ rdl = r24, xl, yl, zl เท่านั้น

```
- mul    rr,rr                ; r1:r0 <-- rr x rr
```

7. คำสั่ง DAA ไม่มี ต้องคำนวณเองนะจ๊ะ

œœœœœœœœœœ