Notas ensamblador AVR

Resumen de algunas notas sobre expresiones, directivas y convenios de ensamblador del AVR.

Expresiones

Constantes enteras

- Decimal: 10, 255.
- Hexadecimal (0x, $): 0x0a, $0a, 0xff, $ff.
- Binario (0b): 0b00001010, 0b11111111.
- Octal(0): 010, 077.

Operadores

Actúan sobre constantes.

- |: NOT lógico.
- ~: NOT bit a bit.
- -: menos unario
- + - * /: aritméticos
- & | ^: lógicos bit a bit (AND, OR, XOR).

Funciones

- *low(x)*: byte menos significativo.
- *high(x)*: byte más significativo.
- *exp2(x)*: 2 elevado a x.
- *log2(x)*: parte entera del logaritmo en base 2 de x.
- *abs(x)*: valor absoluto.

Directivas

- *.cseg*: segmento de código
- *.dseg*: segmento de datos
- *.eseg*: segmento EEPROM.
- *.org*: establece el contador de direcciones (programa, datos o eeprom).

```
.dseg ; segmento de datos
.org 0x100
...
```

```
.cseg ; segmento de código
.org 0x20
...
.eseg ; segmento EEPROM
...
```

- *.byte*: reserva bytes en la memoria de datos.
- *.db*: define bytes en la memoria de programa o eeprom.
- *.dw*: define palabras (16 bits) en la memoria de programa o eeprom.

```
.dseg
var:    .byte 1 ; variable de 1 byte.
list:   .byte 12 ; lista de 12 bytes.
...
.cseg
data:   .db 7, 0xff, 0b10101111
wdata:  .dw 13500, 0xf8a9
```

- *.def*: define un nombre simbólico par un registro.

```
.def counter=r16
.cseg
ldi counter,16
...
```

- *.equ*: asigna un valor a una etiqueta.

```
.equ porta = 0x25
.cseg
out porta,r16
```

- *.include*: incluye otro archivo.

Uso de registros

Convenio de uso de registros para compatibilidad con otros programas y con el compilador de C.

Recuerda: los registros *r0* a *r15* no permiten direccionamiento inmediato (*ldi*, etc.).

- **r0**: registro temporal. Puede ser alterado por rutinas en C pero es preservado por las rutinas de interrupción. Suele usarse para cálculos temporales y su contenido puede desecharse una vez completado el cálculo.
- **r1**: se asume que siempre vale 0. Puede usarse temporalmente, pero luego debe volver a ponerse a cero (`clr r1`).

- **r18-r27, r30-r31**: usados en subrutinas (*call*). Pueden usarse libremente en el programa principal, pero la llamada a una subrutina puede alterar su valor. El *llamador* es responsable de salvar su valor (*push*) si fuera necesario.
- **r2-r17, r28-r29**: preservados por las subrutinas. Pueden usarse libremente. No es necesario preservar su valor antes de la llamada a una subrutina, ya que la misma hará la tarea si es necesario.

Consecuencias:

- Los registros *r16*, *r17* e *y*(*r29:r28*) son especialmente convenientes en el programa principal: pueden usarse con cualquier tipo de direccionamiento y no serán modificados por las subrutinas.
- En el programa principal, además, es ventajoso usar los registros *r2-r15* salvo que se necesite operar con datos inmediatos.
- Si el programa principal usa los registros *r18-r25*, *x*(*r27:r26*) o *z*(*r31:r30*) deberá guardar sus valores en la pila antes de llamar a cualquier subrutina y recuperar su valor posteriormente.
- En una subrutina, es conveniente usar *r18-r25*, *x*(*r27:r26*) y *z*(*r31:r30*) ya que no es necesario restaurar su valor antes de regresar de la subrutina (*ret*).
- Si una subrutina desea utilizar los registros *r2-r15* o *y*(*r29:r28*) deberá guardar sus valores en la pila al comenzar la subrutina y recuperar su valor antes de volver al programa principal.