AVR Memory

M. Anwar Ma’sum, Bayu Anggorojati, Grafika Jati

Contoh Program

```
.include “m8515def.inc“

forever:

   ldi R26, 04 ; 04 diartikan sebagai nilai desimal 4

   ldi R27, $02 ; $02 diartikan sebagai 02 dalam hexadecimal

   ld R16, X

   andi R16, 0x63

   st X, R16

   rjmp forever
```

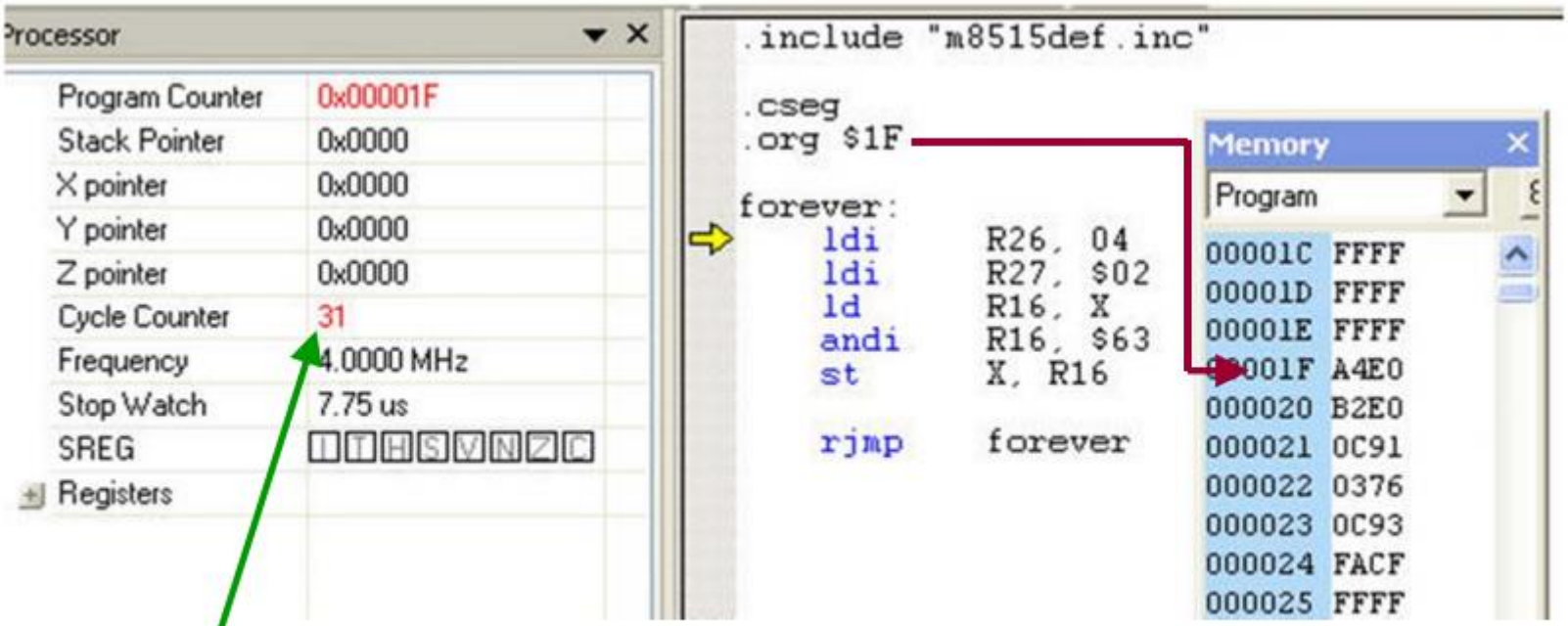
0x02 = $02

Directive – code segment

- Beberapa Directive yang sering digunakan:

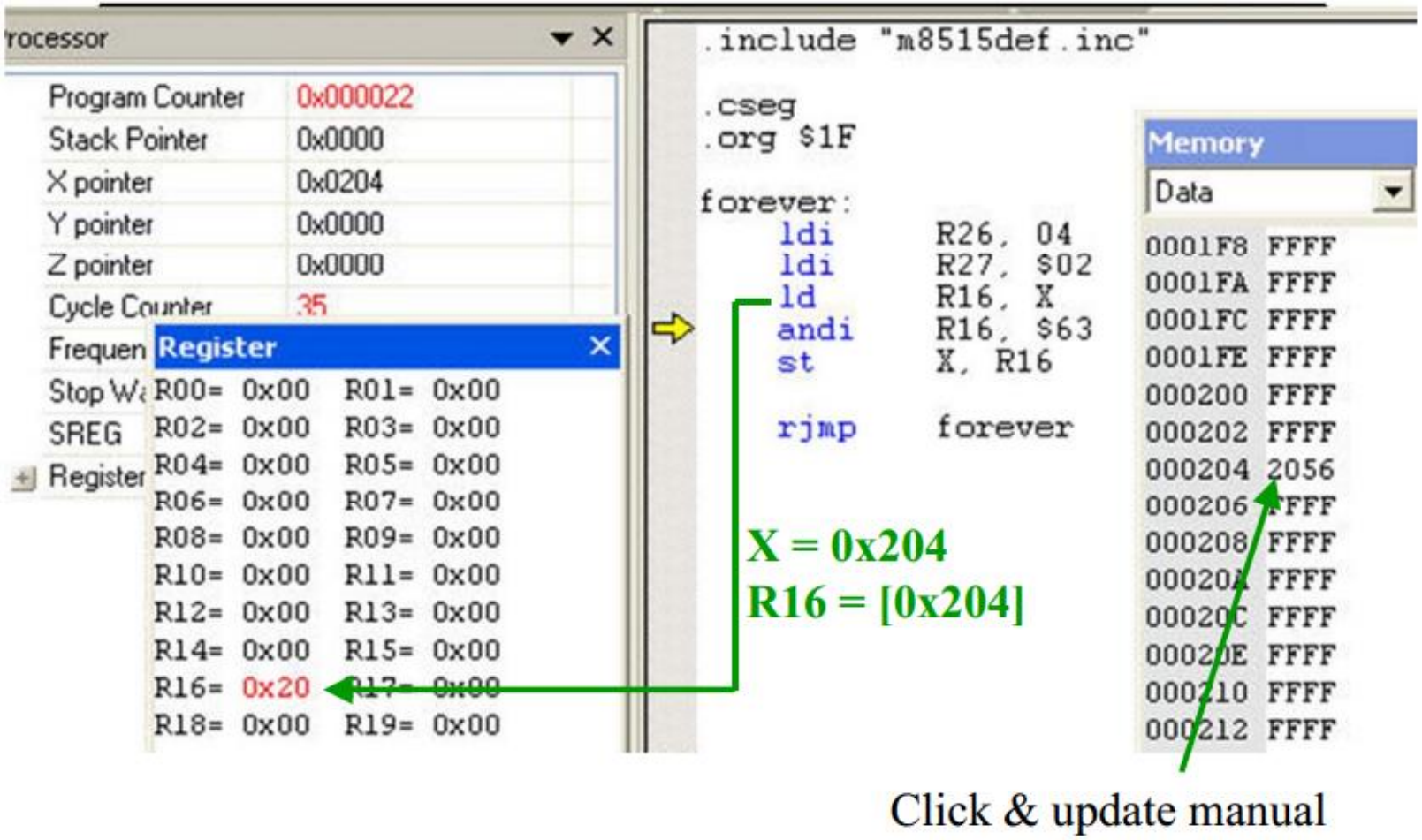
Directive	Fungsi	Contoh
.include	Memasukan definisi-definisi dari tipe prosesor yang digunakan.	.include "8515def.inc"
.def	Mendeskripsikan nama dari register	.def temp2 = r17
.equ	Mendeskripsikan sebuah nilai konstanta	. equ CONS = 123
.db	Mendefinisikan nilai yag akan disimpan pada program memory	.db "1,2"
.macro & .endmacro	Menandai dimulai & selesainya MACRO	.macro NAMAMACRO ..... .endmacro
.org	Mendefinisikan alamat penyimpanan baris kode pada alamat program memory tertentu	

Directive – code segment (cont.)

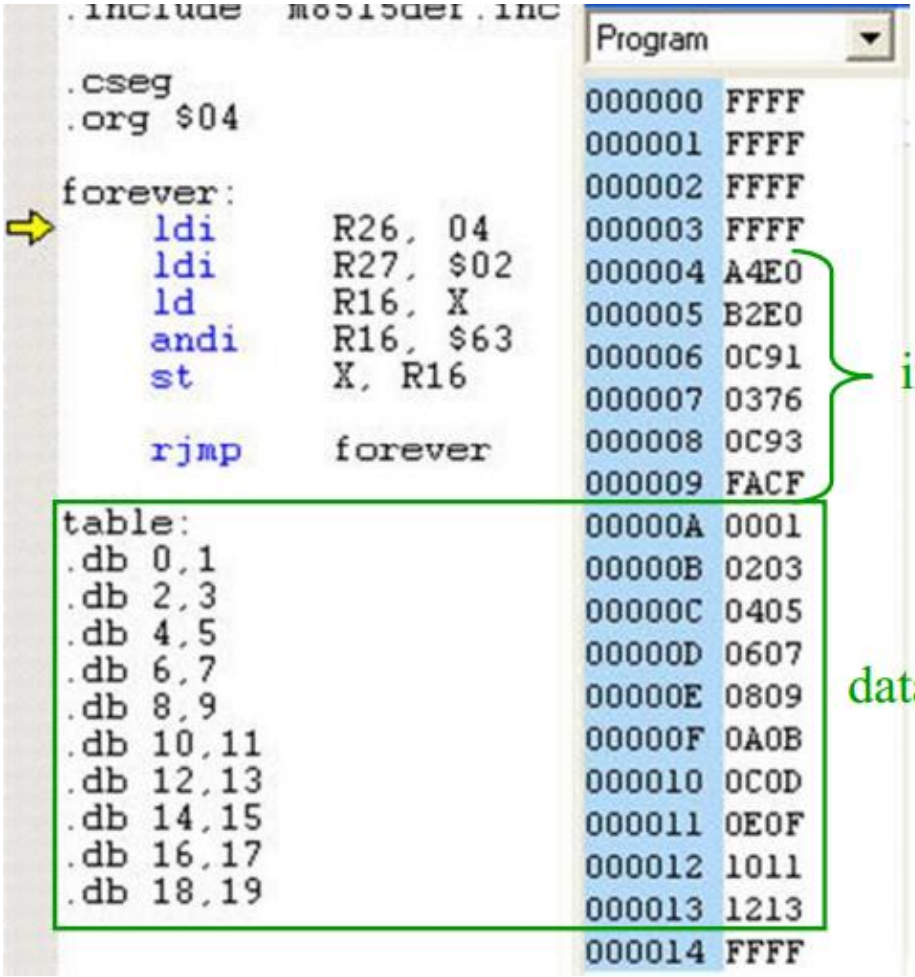


31 clock needed to reach 1st instruction

Directive – code segment (cont.)



Directive – define bytes

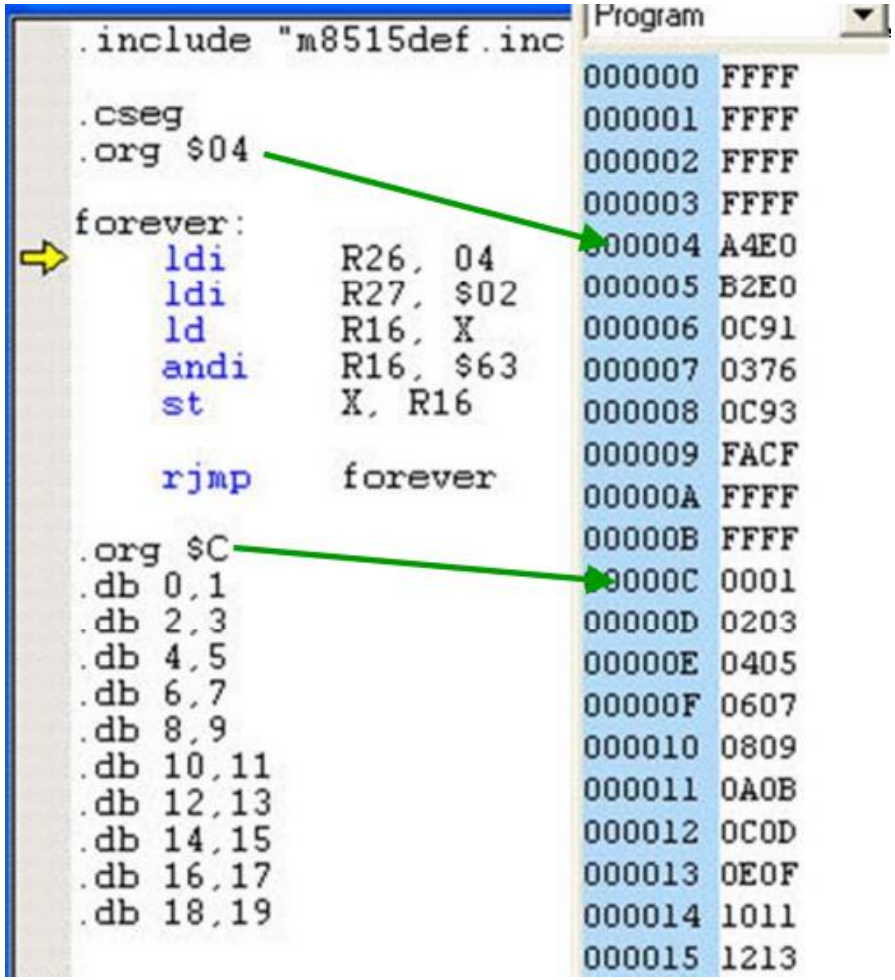


is directive for putting data into program memory (flash).

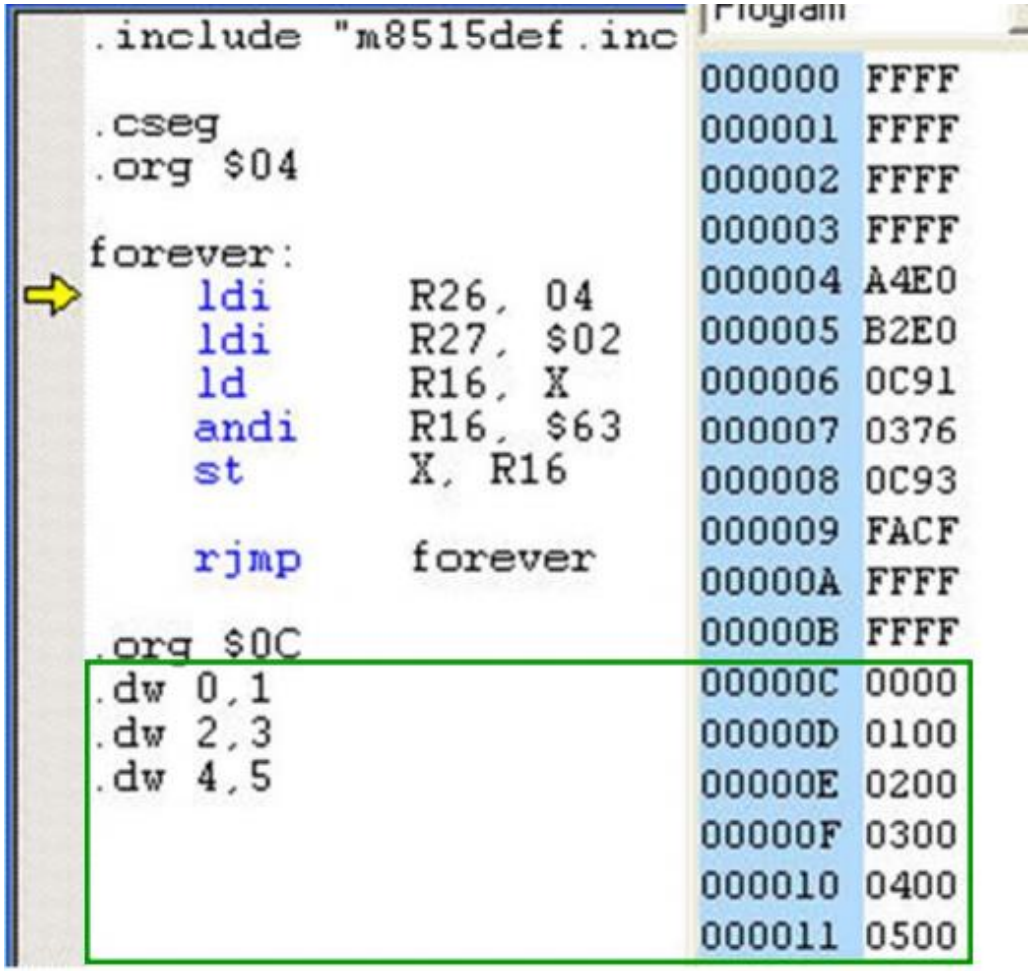
instructions

data in program memory

Directive – origin



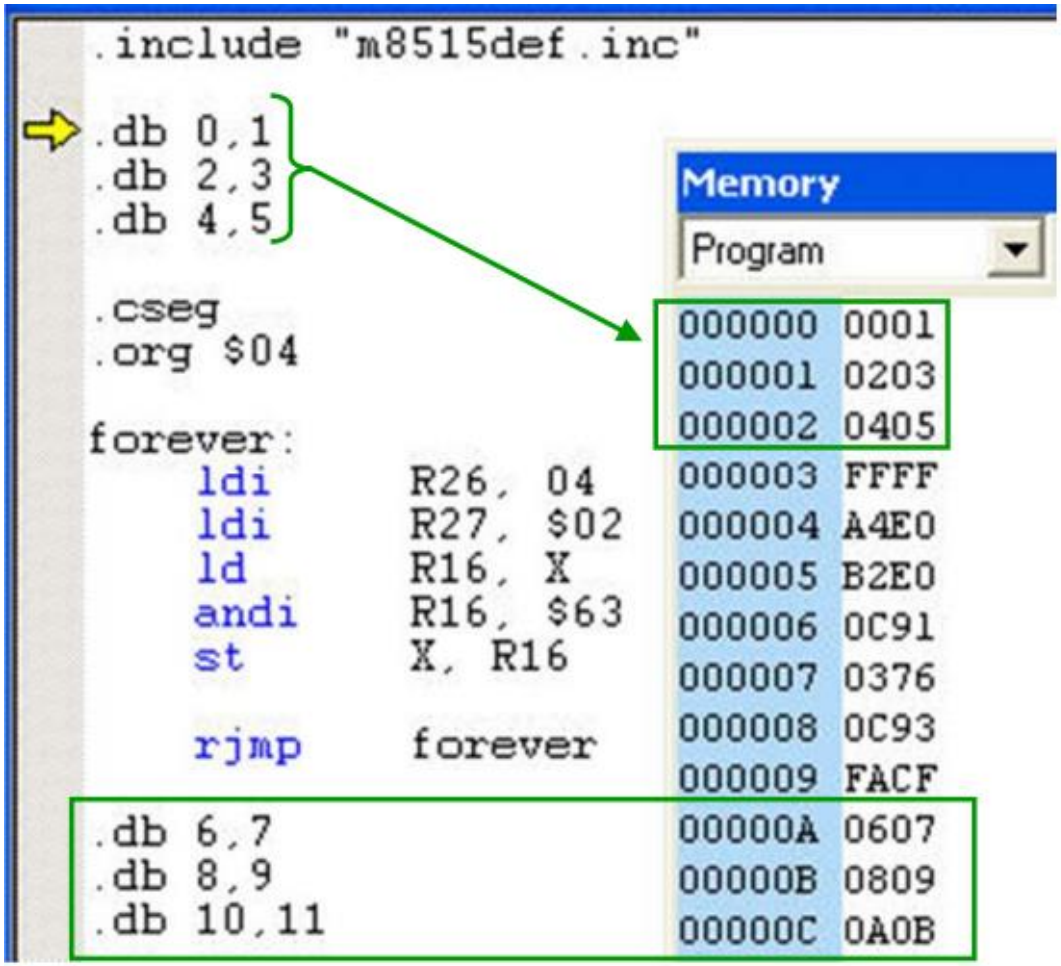
Directive – define words



```
.include "m8515def.inc

.cseg
.org $04

forever:
    ldi     R26, 04
    ldi     R27, $02
    ld      R16, X
    andi    R16, $63
    st      X, R16

    rjmp    forever

.org $0C
.dw 0,1
.dw 2,3
.dw 4,5
```

```
Program
000000 FFFF
000001 FFFF
000002 FFFF
000003 FFFF
000004 A4E0
000005 B2E0
000006 0C91
000007 0376
000008 0C93
000009 FACF
00000A FFFF
00000B FFFF
00000C 0000
00000D 0100
00000E 0200
00000F 0300
000010 0400
000011 0500
```

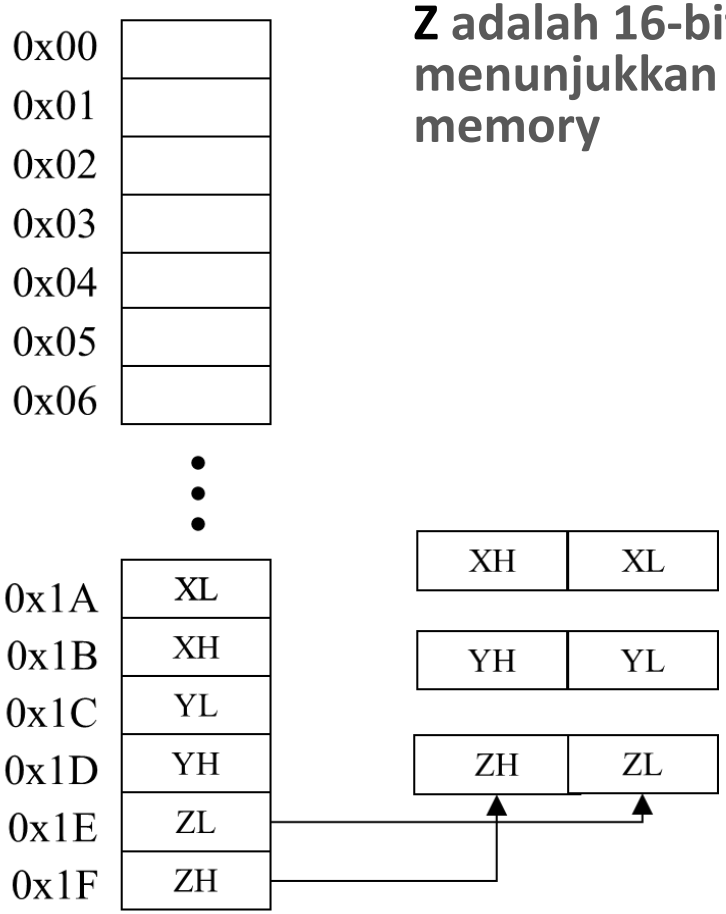
Directive – .db dengan letak berbeda

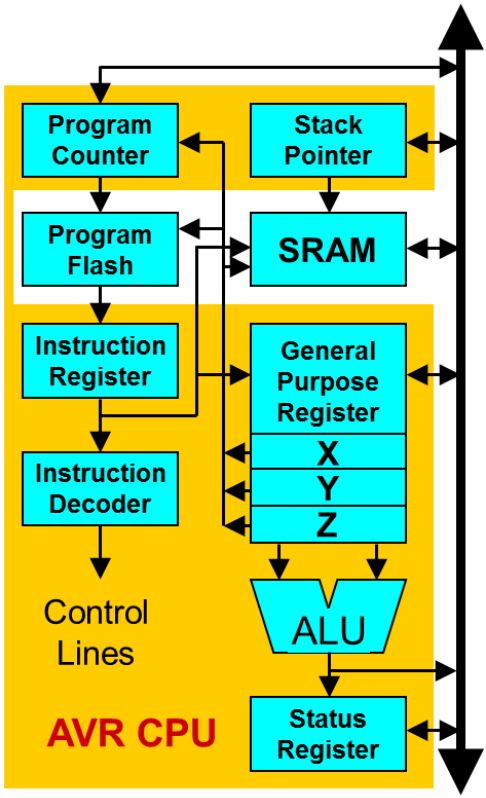


AVR Register: X, Y, Z

X, Y, Z adalah 16-bit pointer register, digunakan untuk menunjukkan alamat dengan max 16-bit pada SRAM

Z adalah 16-bit pointer register, dapat digunakan untuk menunjukkan alamat dengan max 16-bit pada program memory





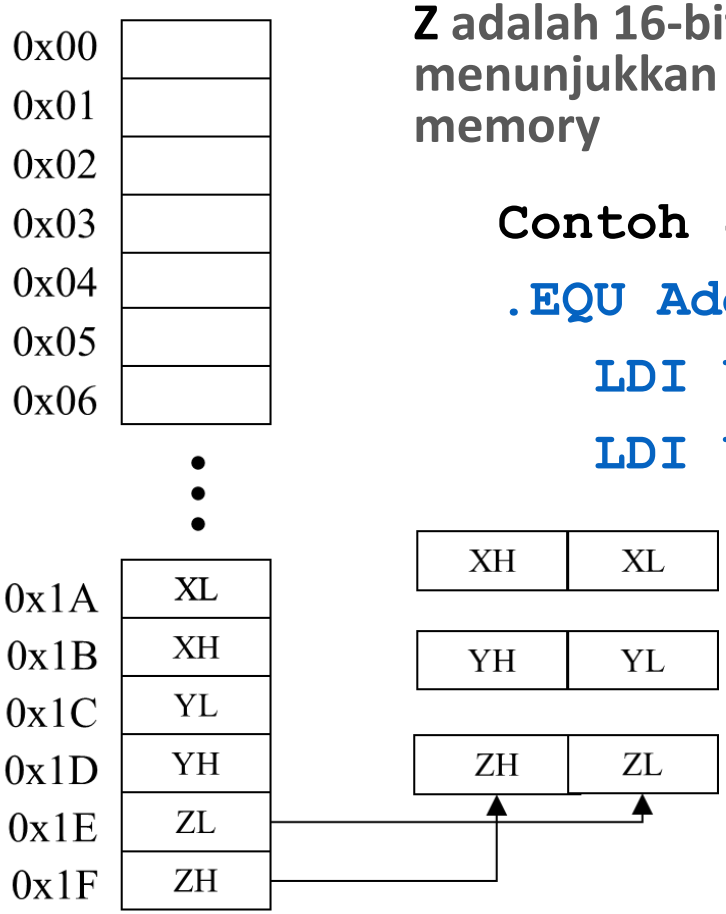
AVR Register: X, Y, Z

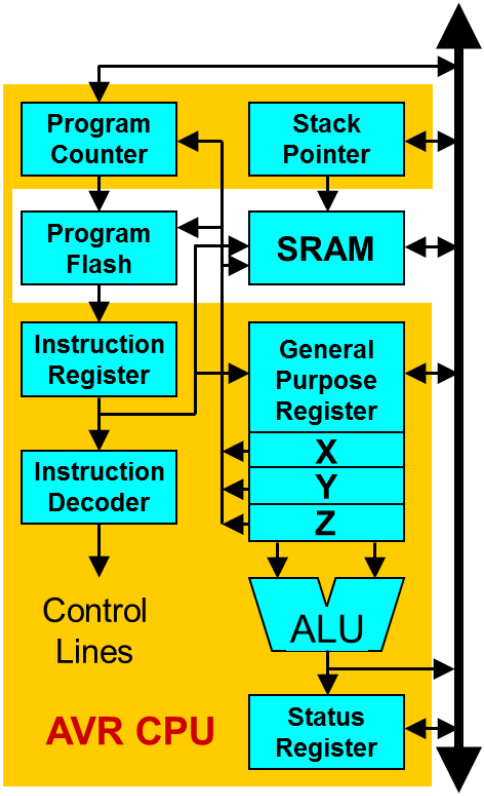
X, Y, Z adalah 16-bit pointer register, digunakan untuk menunjukkan alamat dengan max 16-bit pada SRAM

Z adalah 16-bit pointer register, dapat digunakan untuk menunjukkan alamat dengan max 16-bit pada program memory

Contoh 4:

```
.EQU Address = RAMEND
    LDI YH, HIGH(Address)
    LDI YL, LOW(Address)
```



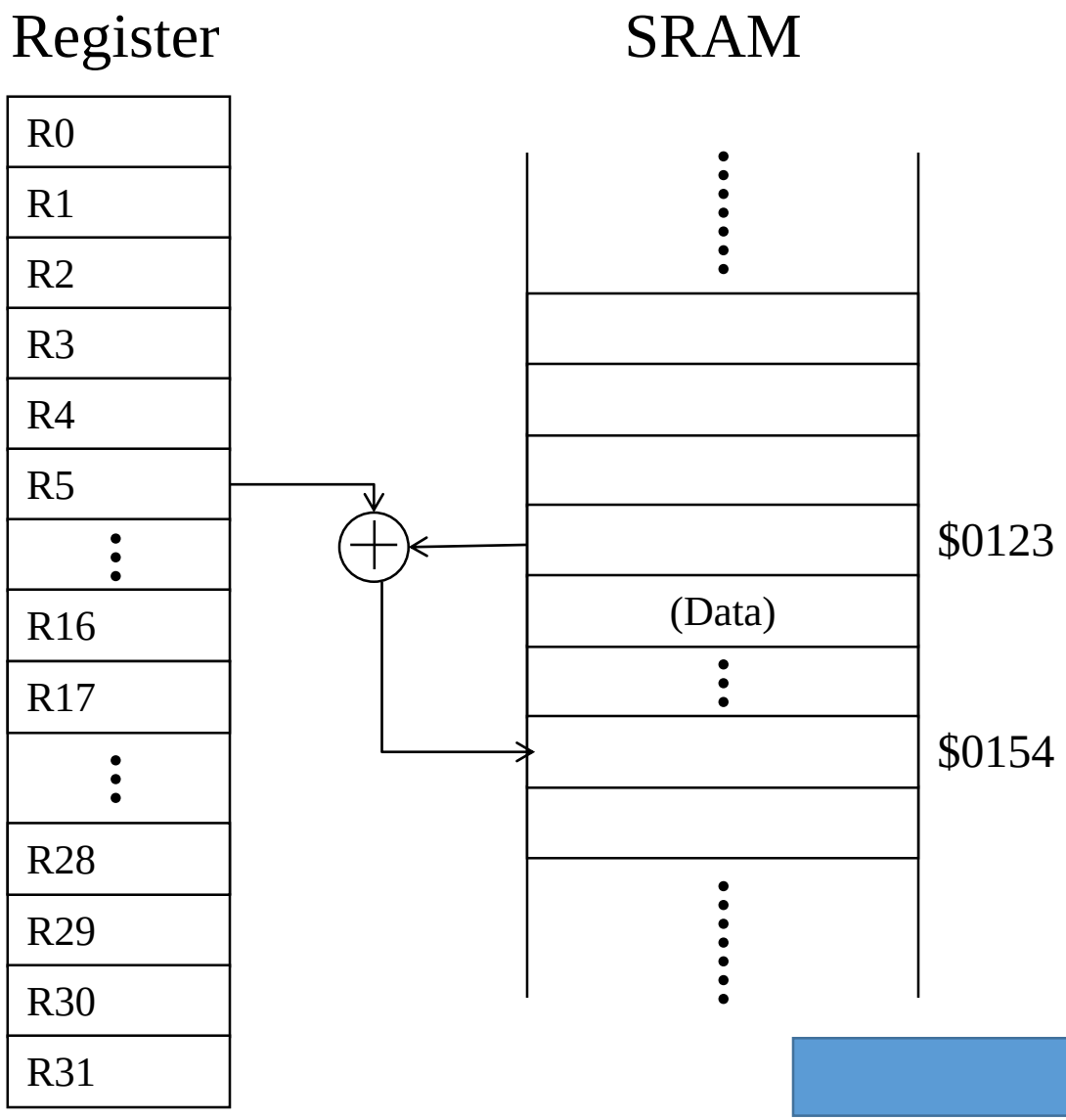


AVR Register

Pointer	Sequence	Examples
X	Read/Write dari/ke alamat X, jangan nilai ubah pointernya	LD R1,X ST X,R1
X+	Read/Write dari/ke alamat X dan setelah itu increment satu nilai pointernya	LD R1,X+ ST X+,R1
-X	Decrement satu nilai pointer dan kemudian read/write dari/ke alamat yang baru	LD R1,-X ST -X,R1

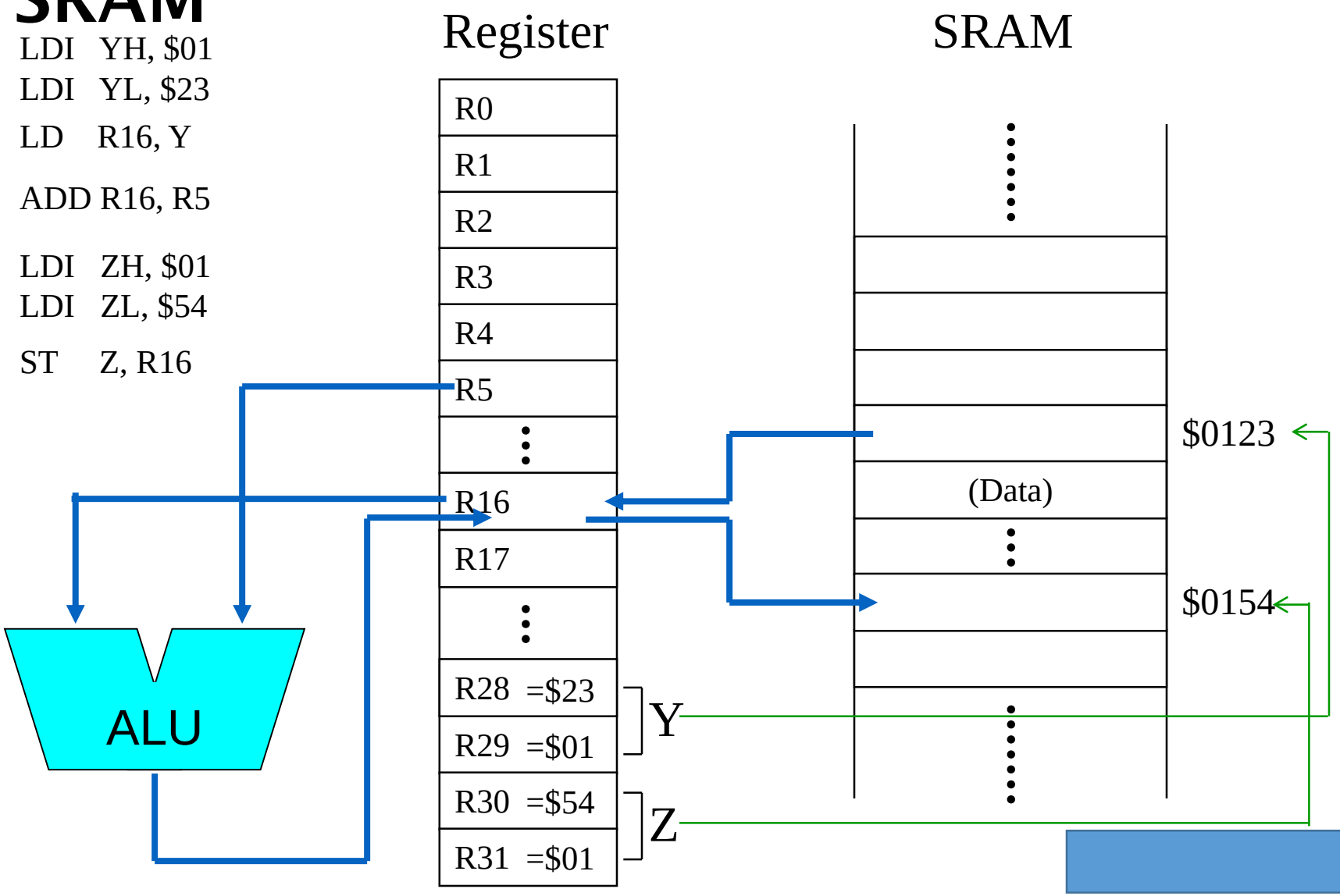
Menggunakan Data dari SRAM

Jumlahkan isi R5 dengan data di memory address $0123. Simpan hasilnya di memory address $0154.

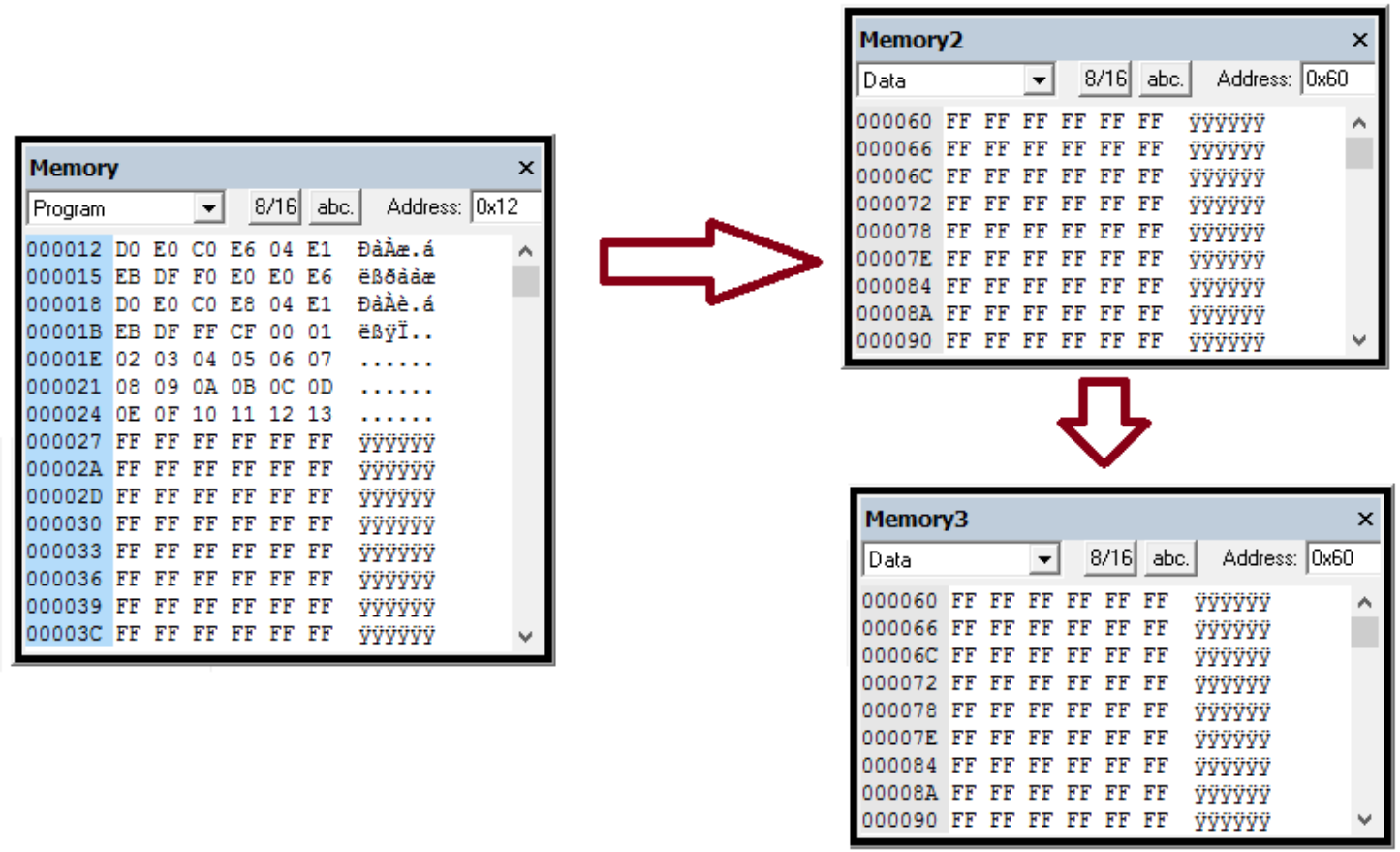


Menggunakan Data dari SRAM

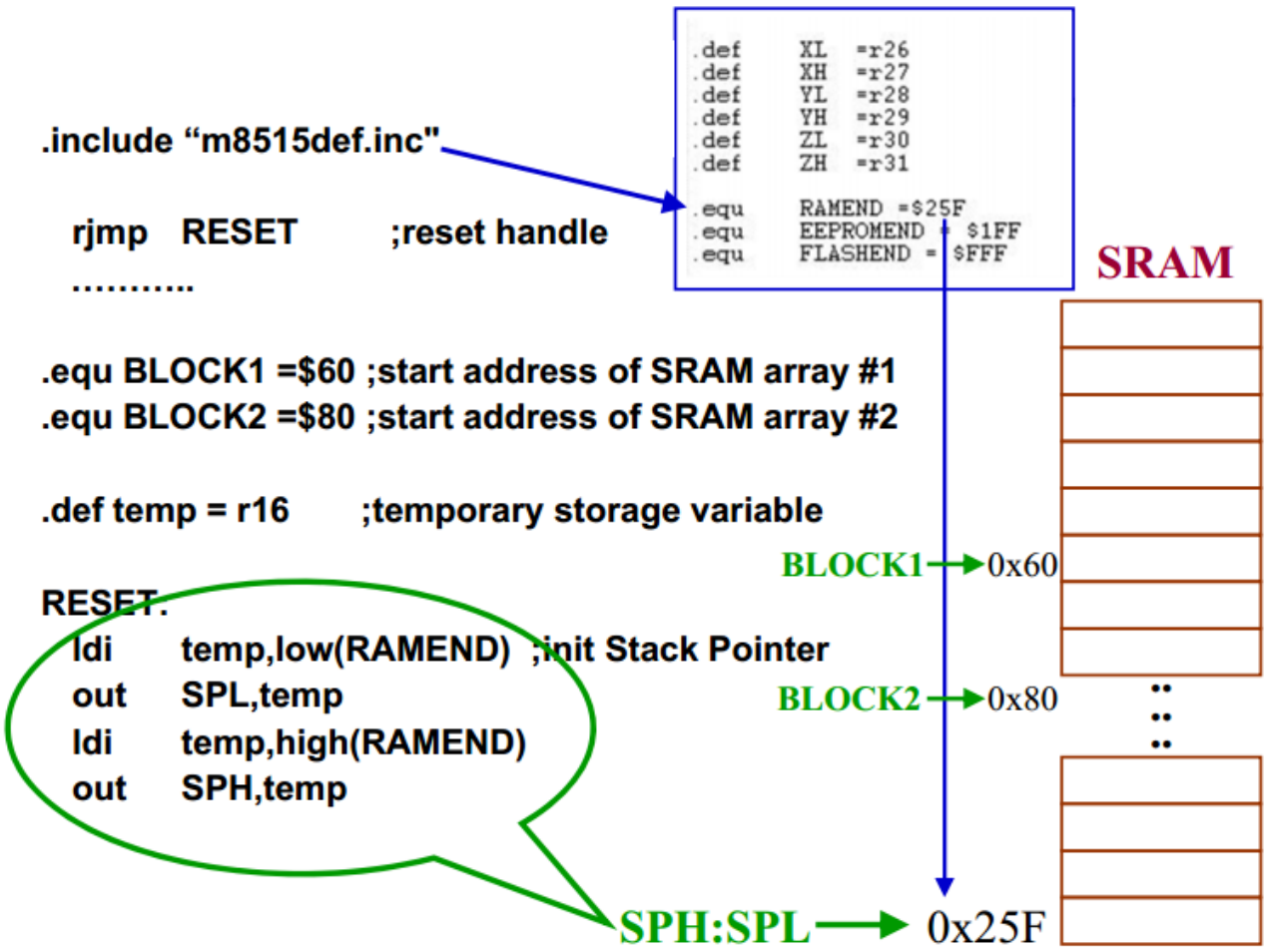
```
LDI   YH, $01
LDI   YL, $23
LD    R16, Y

ADD R16, R5

LDI   ZH, $01
LDI   ZL, $54

ST    Z, R16
```



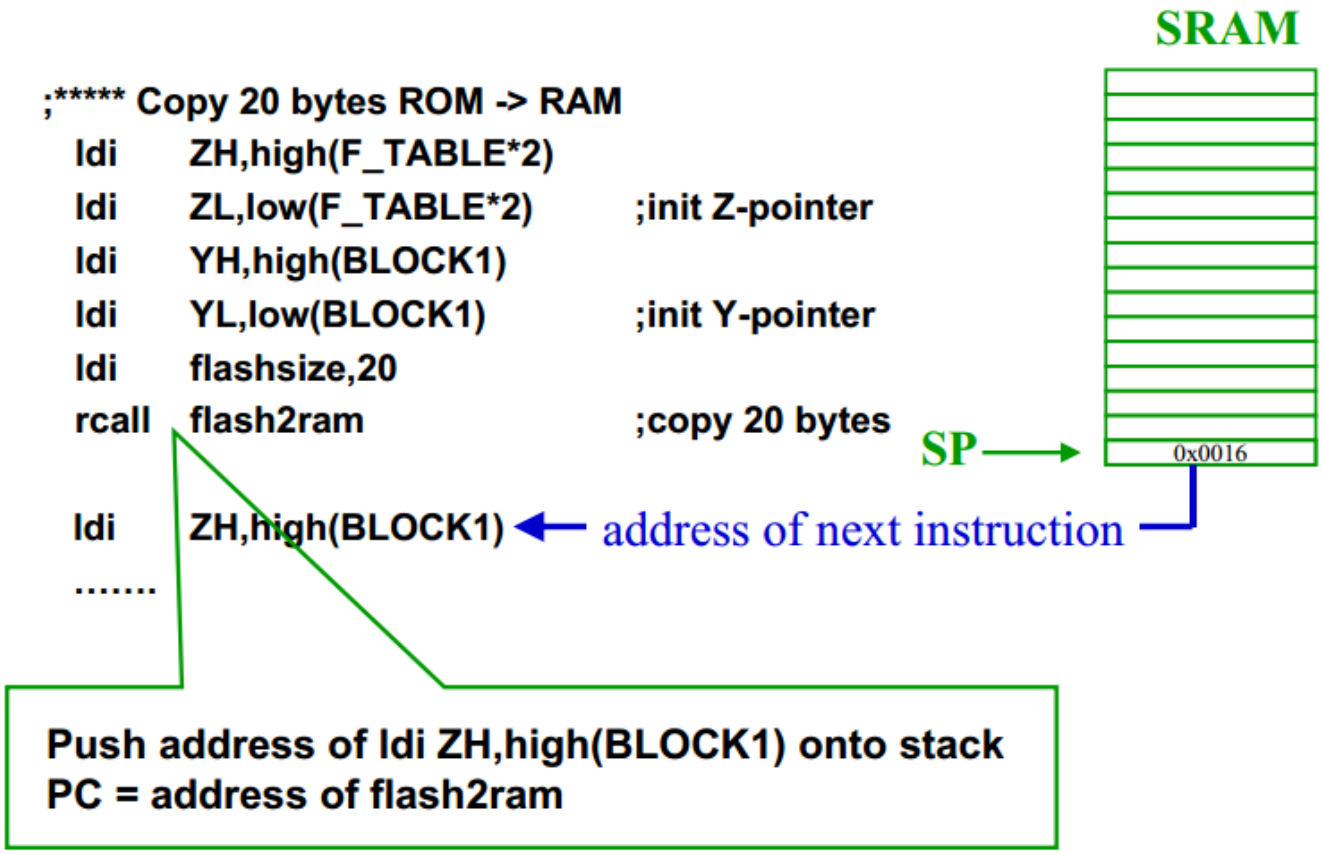
Contoh program: avr102.asm



Init Stack Pointer



Copy Program Memory to Data Memory



Copy Program Memory to Data Memory (cont.)

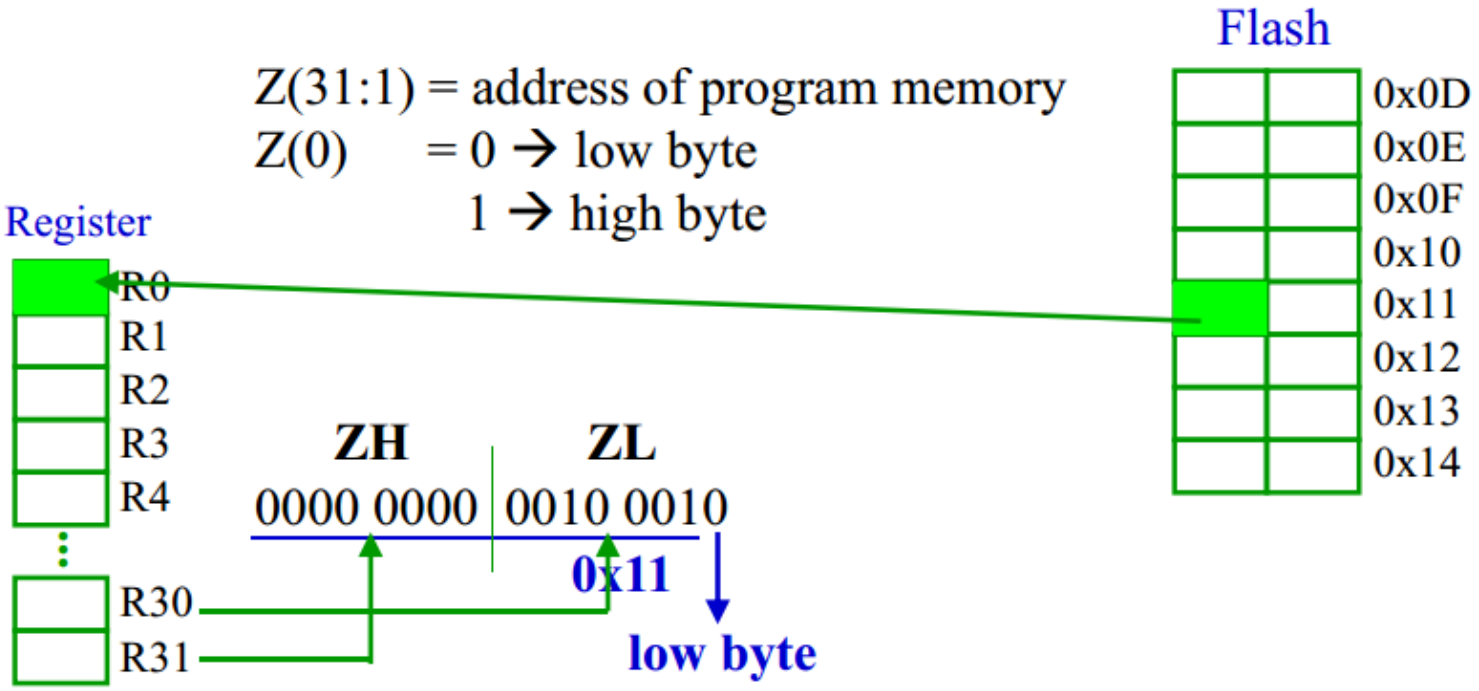
```
;***** Subroutine Register variables
.def    flashsize=r16   ;size of block to be copied

flash2ram:
  lpm                   ;get constant
  st    Y+,r0           ;store in SRAM and increment Y-pointer
  adiw  ZL,1            ;increment Z-pointer
  dec   flashsize
  brne  flash2ram       ;if not end of table, loop more
  ret
```

PC = Pop(stack)
- Copy the value pointed by TOS to PC
- Increment TOS

Copy Data Memory to Data Memory

```
;***** Copy 20 bytes RAM -> RAM
  ldi   ZH,high(BLOCK1)
  ldi   ZL,low(BLOCK1)   ;init Z-pointer
  ldi   YH,high(BLOCK2)  ;
  ldi   YL,low(BLOCK2)   ;init Y-pointer
  ldi   ramsize,20
  rcall ram2ram          ;copy 20 bytes

forever:
  rjmp forever       ;eternal loop
```

Pointer Z ke Program Memory



Low byte of program memory's content at address 0x11 is copied to R0