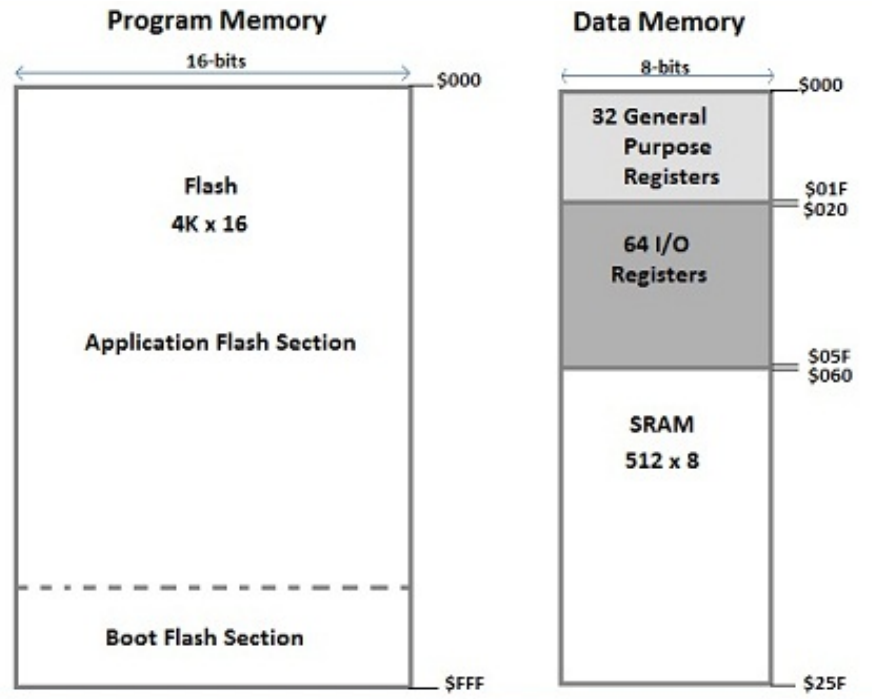
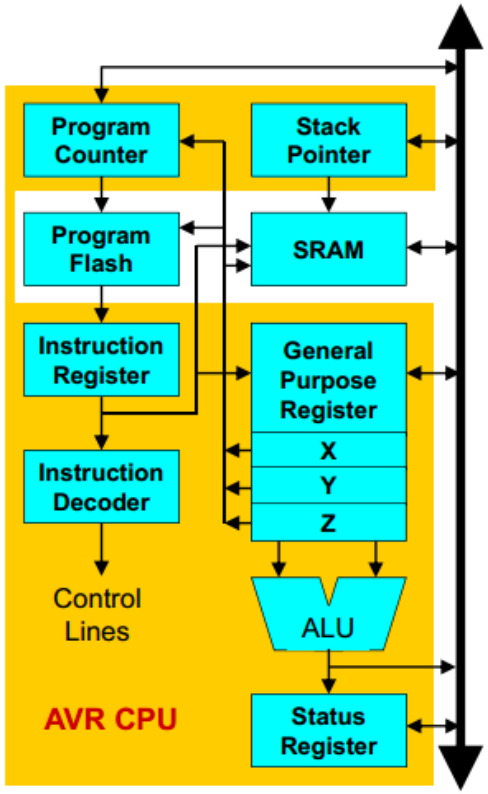
I/O Registers in AVR

Tim Dosen POK

I/O Registers

- Special purpose registers for Input and Output in AVR.





Address and Function of I/O registers

Address	Hex Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU general Control Register
$33 ($53)	TCCR0	Timer/Counter 0 Control Register
$32 ($52)	TCNT0	Timer/Counter 0 (8-bit)
$2F ($4F)	TCCR1A	Timer/Counter 1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter 1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter 1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter 1 Low Byte
$2B ($4B)	OCR1AH	Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Output Compare Register A Low Byte
$29 ($49)	OCR1AH	Output Compare Register B High Byte
$28 ($48)	OCR1AL	Output Compare Register B Low Byte
$25 ($45)	ICR1H	T/C 1 Input Capture Register High Byte
$24 ($44)	ICR1L	T/C 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register

Address and Function of I/O registers

Address	Hex Name	Function
$1B ($38)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$15 ($35)	PORTC	Data Register, Port C
$14 ($34)	DDRC	Data Direction Register, Port C
$13 ($33)	PINC	Input Pins, Port C
$12 ($32)	PORTD	Data Register, Port D
$11 ($31) DDRD	Data Direction Register, Port D	
$10 ($30)	PIND	Input Pins, Port D
$0F ($2F)	SPDR	SPI I/O Data Register
$0E ($2E)	SPSR	SPI I/O Status Register
$0D ($2D)	SPCR	SPI I/O Control Register
$0C ($2C)	UDR	UART I/O Data Register
$0B ($2B)	USR	UART Status Register
$0A ($2A)	UCR	UART Control Register
$09 ($29)	UBRR	UART Baud Rate Register
$08 ($28)	ACSR	Analog Comparator Control and Status Register

*Deskripsi lebih lengkap baca AVR IO Registers Application Note!

Instructions for I/O

- P: I/O Register
- Data Transfer:
  - IN Rd, P ; Rd ← P
  - OUT P, Rs ; P ← Rs
- Bit Operation:
  - SBI P, b ; P(b) ← 1, Bit # b of register P is set to “1”
  - CBI P, b ; P(b) ← 0, Bit # b of register P is set to “0”
- Special:
  - SEI ; Global Interrupt enable
  - CLI ; Global Interrupt disable

IN - Load an I/O Location to Register

Description:

Loads data from the I/O Space (Ports, Timers, Configuration Registers etc.) into register Rd in the Register File.

Operation:

(i) Rd ← I/O(A)

	Syntax:	Operands:	Program Counter:
(i)	IN Rd,A	$0 \le d \le 31$, $0 \le A \le 63$	PC ← PC + 1

16-bit Opcode:

1011	0AAd	dddd	AAAA

OUT – Store Register to I/O Location

Description:

Stores data from register Rr in the Register File to I/O Space (Ports, Timers, Configuration Registers etc.).

Operation:

(i) I/O(A) ← Rr

	Syntax:	Operands:	Program Counter:
(i)	OUT A,Rr	$0 \le r \le 31$, $0 \le A \le 63$	PC ← PC + 1

16-bit Opcode:

1011	1AAr	rrrr	AAAA

SBI – Set Bit in I/O Register

Description:

Sets a specified bit in an I/O Register. This instruction operates on the lower 32 I/O Registers – addresses 0-31.

Operation:

(i) I/O(A,b) ← 1

	Syntax:	Operands:	Program Counter:
(i)	SBI A,b	$0 \le A \le 31, 0 \le b \le 7$	PC ← PC + 1

16-bit Opcode:

1001	1010	AAAA	Abbb

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	–	–

Example:

```
out   $1E,r0    ; Write EEPROM address
sbi   $1C,0     ; Set read bit in EECR
in    r1,$1D    ; Read EEPROM data
```

CBI – Clear Bit in I/O Register

Description:

Clears a specified bit in an I/O Register. This instruction operates on the lower 32 I/O Registers – addresses 0-31.

	Operation:
(i)	I/O(A,b) ← 0

	Syntax:	Operands:	Program Counter:
(i)	CBI A,b	$0 \le A \le 31, 0 \le b \le 7$	PC ← PC + 1

16-bit Opcode:

1001	1000	AAAA	Abbb

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	–	–	–	–	–

Example:

```
cbi     $12,7          ; Clear bit 7 in Port D
```

SEI – Set Global Interrupt Flag

Description:

Sets the Global Interrupt Flag (I) in SREG (Status Register). The instruction following SEI will be executed before any pending interrupts.

Operation:

(i) I ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEI	None	PC ← PC + 1

16-bit Opcode:

1001	0100	0111	1000

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
1	–	–	–	–	–	–	–

I: 1
Global Interrupt Flag set

Example:

```
sei             ; set global interrupt enable
sleep           ; enter sleep, waiting for interrupt
                ; note: will enter sleep before any pending interrupt(s)
```

CLI – Clear Global Interrupt Flag

Description:

Clears the Global Interrupt Flag (I) in SREG (Status Register). The interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction.

Operation:

(i) I ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLI	None	PC ← PC + 1

16-bit Opcode:

1001	0100	1111	1000

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
0	–	–	–	–	–	–	–

I: 0
Global Interrupt Flag cleared

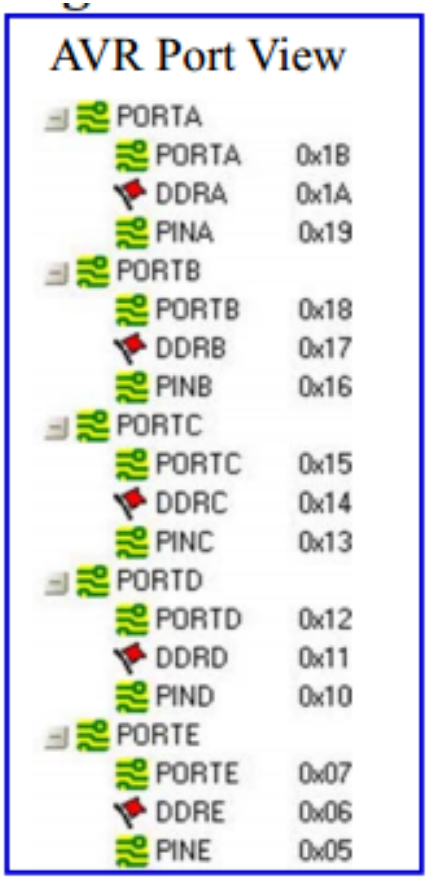
Example:

```
in      temp, SREG  ; Store SREG value (temp must be defined by user)
cli                 ; Disable interrupts during timed sequence
sbi     EECR, EEMWE ; Start EEPROM write
sbi     EECR, EEWE
out     SREG, temp  ; Restore SREG value (I-Flag)
```

I/O Ports

- Port is used to connect CPU with external components.

```
.include "m8515def.inc"
.equ    EEDR    =$1d
.equ    EECR    =$1c
.equ    PORTA   =$1b
.equ    DDRA    =$1a
.equ    PINA    =$19
.equ    PORTB   =$18
.equ    DDRB    =$17
.equ    PINB    =$16
.equ    PORTC   =$15
.equ    DDRC    =$14
.equ    PINC    =$13
.equ    PORTD   =$12
.equ    DDRD    =$11
.equ    PIND    =$10
```



I/O Ports (cont.)

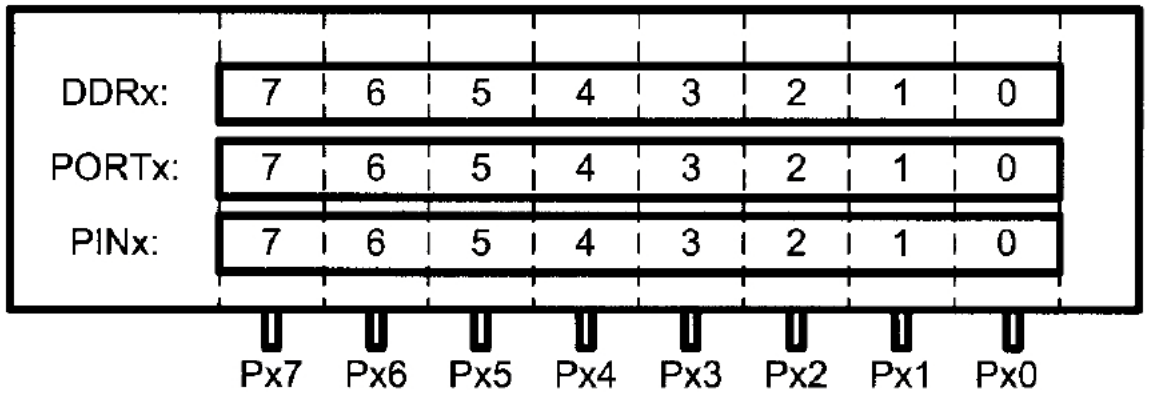
Each port has three I/O registers associated with it.

DDRx = Data Direction Register
PORTx = Data Register
PINx = Port Input Pins

http://maxembedded.files.wordpress.com/2011/06/different-port-operations.png

I/O Ports (cont.)

- I/O Registers are 8-bit wide
- Each port has a maximum of 8 pins
- Relations between the Registers and the Pins of AVR



I/O Ports (cont.)

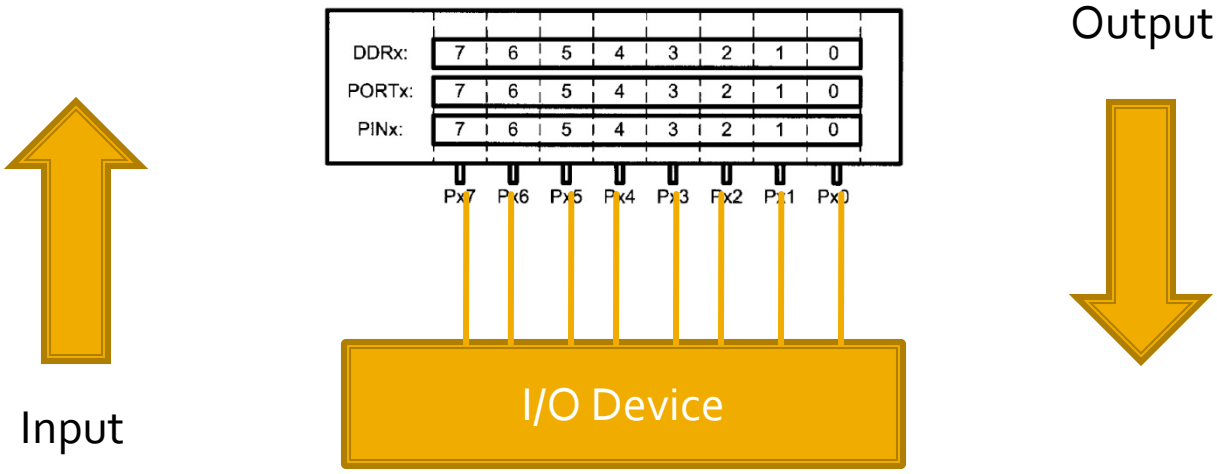
- Any port can be used as an input or output port
- DDRx I/O register is used to make a given port as an input or output port
  - Input : bit DDR 0
  - Output : bit DDR 1

Bit No.	7	6	5	4	3	2	1	0
Name	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
Initial Value	0	0	0	0	0	0	0	0

Higher Nible

Lower Nible

I/O Ports (cont.)



I/O Ports (cont.)

- For example:
  - To make all Port A pins as input port:
    - DDRA = obooooooooo
  - To make all Port B pins as output port:
    - DDRB = ob11111111
  - To make *lower nibble pins of* Port A as output and *higher nible* pins as input:
    - DDRA = obooooo1111;

I/O Ports (cont.)

```
LDI   R16, oxFF      ; R16 = oxFF = ob 11111111
OUT   DDRB, R16      ; make Port B an output port
LDI   R16, ox55       ; R16 = ox55 = ob 01010101
OUT   PORTB, R16     ; put ox55 on port B pins
```

It must be noted that unless we set the DDRx bits to one, the data will not go from the port register to the pins of AVR. If we remove the first two lines of the above code, the ox55 value will not get to the pins. It will be sitting in the I/O register of Port B inside the CPU