Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

**Politechnika Śląska jako Centrum Nowoczesnego Kształcenia
opartego o badania i innowacje**

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Microprocessor and Embedded Systems

**Faculty of Automatic Control, Electronics and Computer Science,
Informatics, Bachelor Degree**

Lecture 13

AVR-family single-chip microcomputers
Part 1
General architecture

Bartłomiej Zieliński, PhD, DSc

AVR (1)

Program:

(today)

- AVR families
- AVR structure
- Memory organisation, stack
- I/O ports

(next week)

- Built-in peripherials

AVR (1)

- General properties
  - Harvard architecture
    - 16-bit program word
      - 16/32-b instructions, constant command length
    - 8-bit data word
  - 1÷16 MHz clock
  - 8, 20, 28, 40, 84-pin packages
    - In some AVR external data memory possible
    - Some AVR have 8051-compatible pin-out
      - $\overline{\mathrm{EA}}$, $\overline{\mathrm{PSEN}}$ not available
        - » Entire program memory is internal
      - ALE is available
        - » External data memory possible

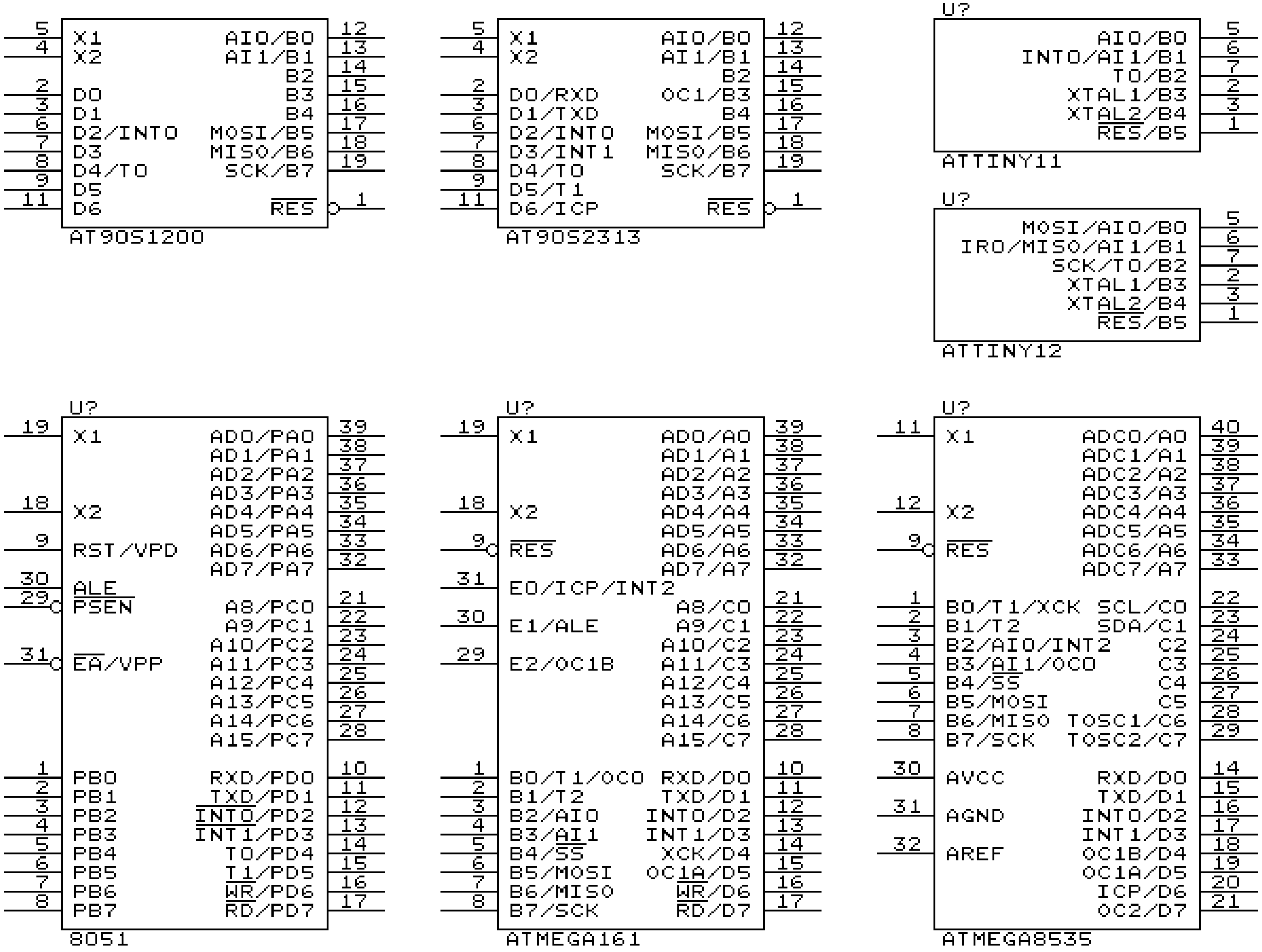
AVR (1)

- AVR families

Family	Flash [K]	EEPROM [B]	SRAM [B]	IO pins	Pins	Int’s	Ext int’s
„basic”	1÷8	128÷512	0÷512	3÷32	8, 20, 28	2÷15	1÷2
Mega AVR	8÷128	512÷4K	512÷4K	32÷54	40, 64	16÷23	2÷17
Tiny AVR	1÷2	0÷256	0÷128	6÷16	8, 20, 28	4÷11	1÷8
Xmega	8-384	1K÷4K	1K÷32K	34÷78	32÷100		

AVR (1)

- AVR pin-out examples



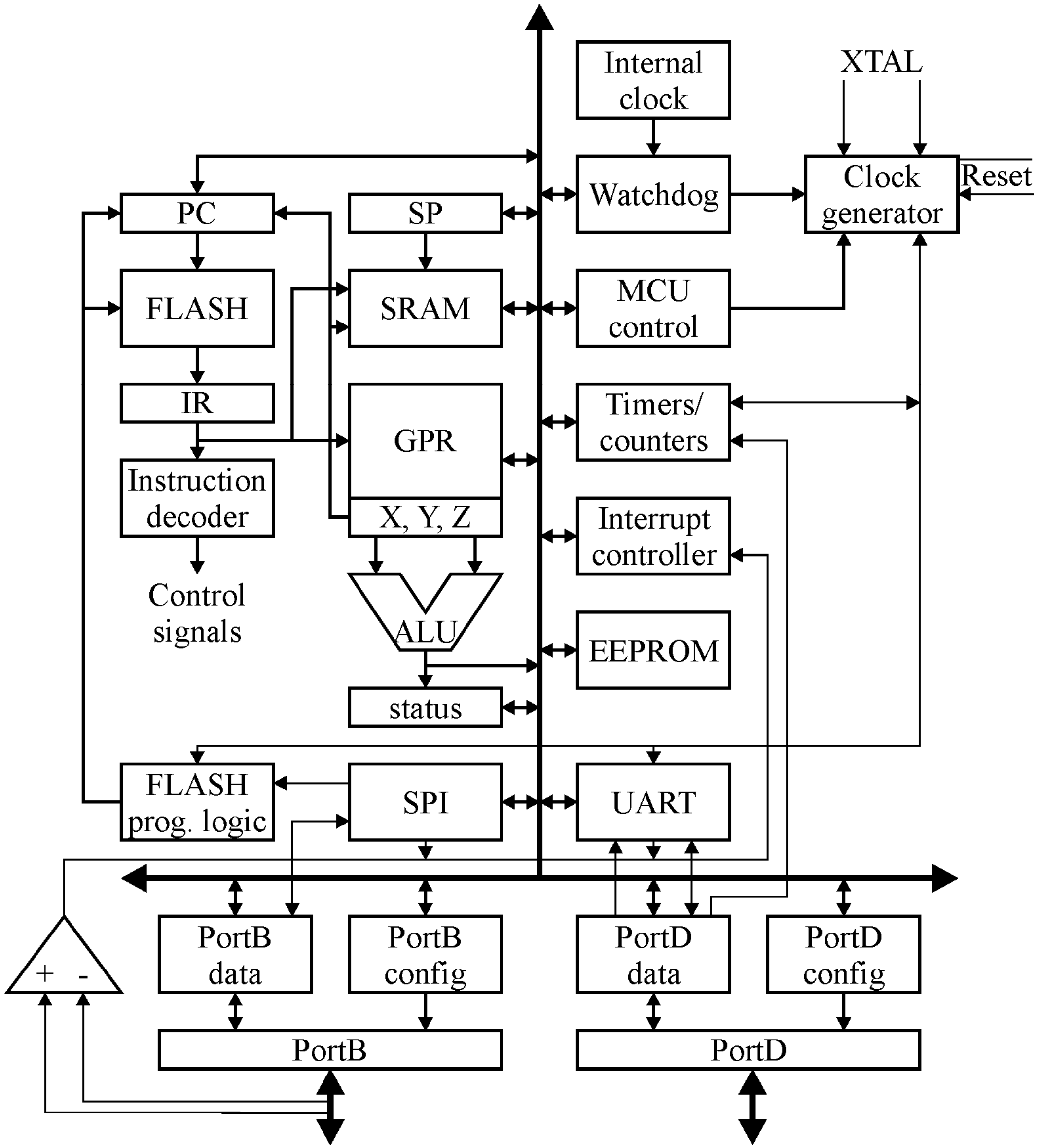
AVR (1)

- AT90S2313 properties
  - 118 commands
  - 2K FLASH, SPI-programmable (1000 cycles)
  - 128 B RAM, 128 B EEPROM (100000 cycles)
  - 32 general purpose registers
  - 15 I/O lines (I_{OL}=I_{OH}=20 mA → direct LED control)
  - V_{CC}=2.7÷6 V (4 MHz ver.), 4÷6 V (10 MHz ver.)
  - Fully static structure (f_{min}=0)
  - Interrupt system incl. external sources
  - 2 power-down modes

AVR (1)

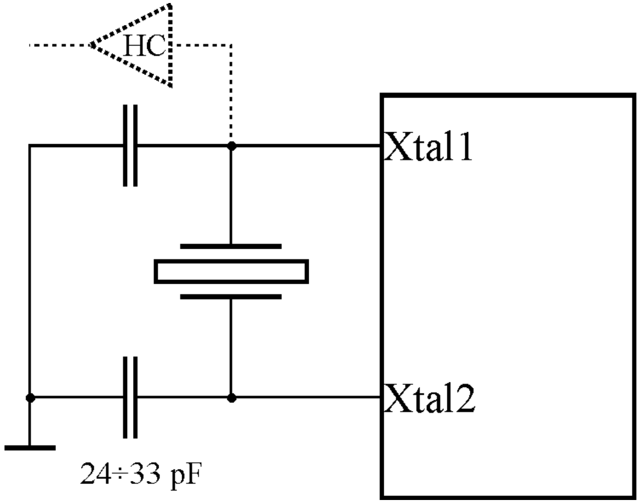
- AT90S2313

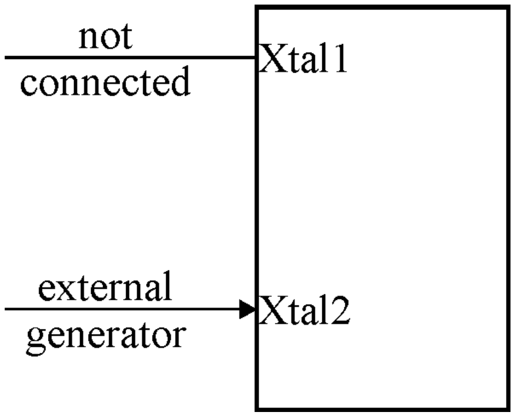
  structure



AVR (1)

- AVR – clock, machine cycles
  - Clock generation
    - System clock

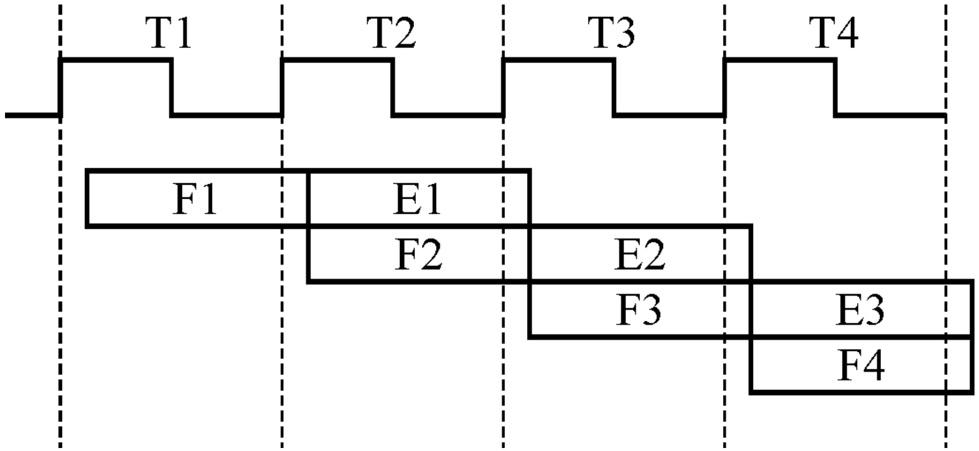




    - Watchdog clock
      - Internal RC circuit, abt. 1 MHz

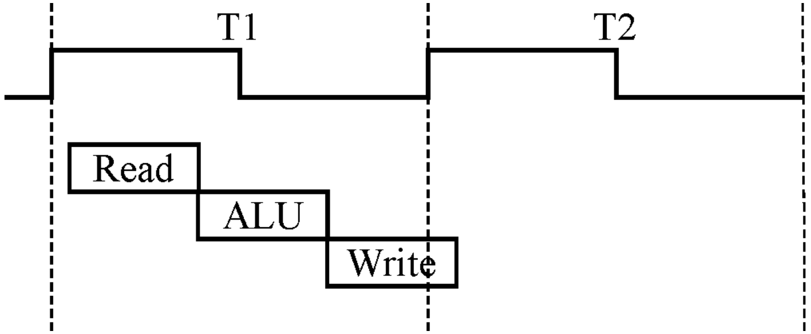
AVR (1)

- AVR – clock, machine cycles
  - Clock frequency not divided
    - Is it multiplied inside?
      - Signals time shift
      - High operation speed
      - But no official information

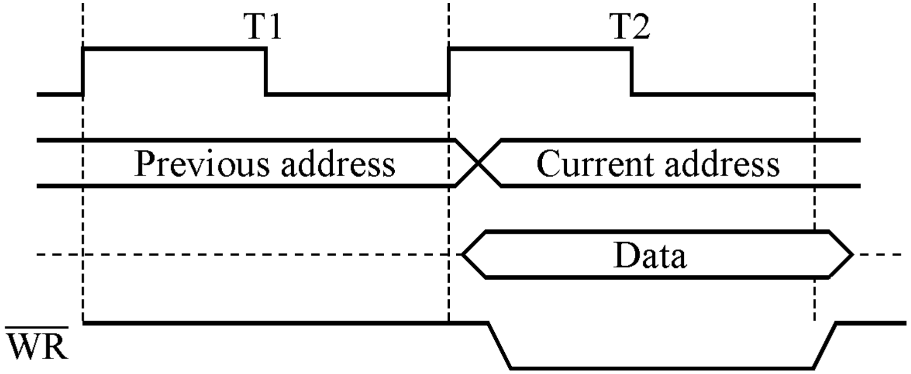


AVR (1)

- AVR – clock, machine cycles
  - Command performed in a single cycle

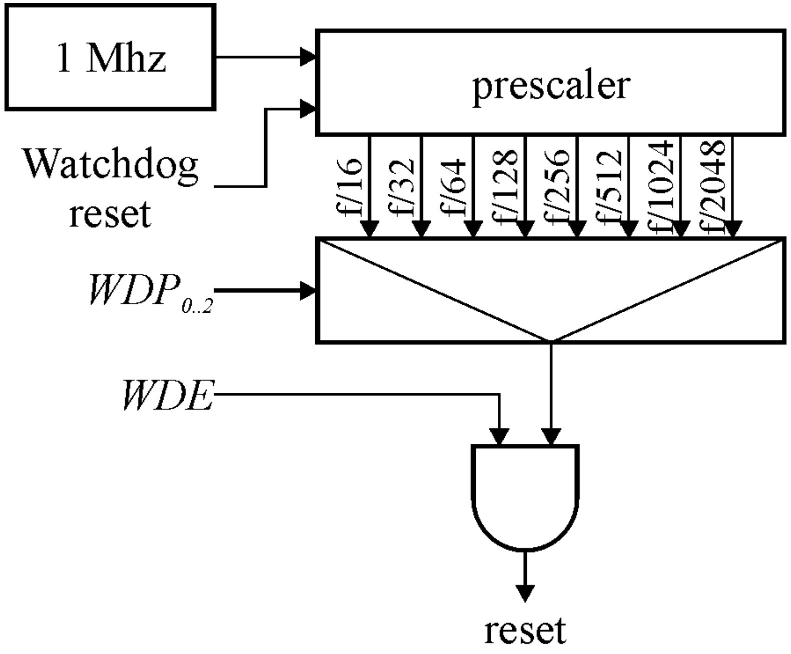


  - Command with internal SRAM access



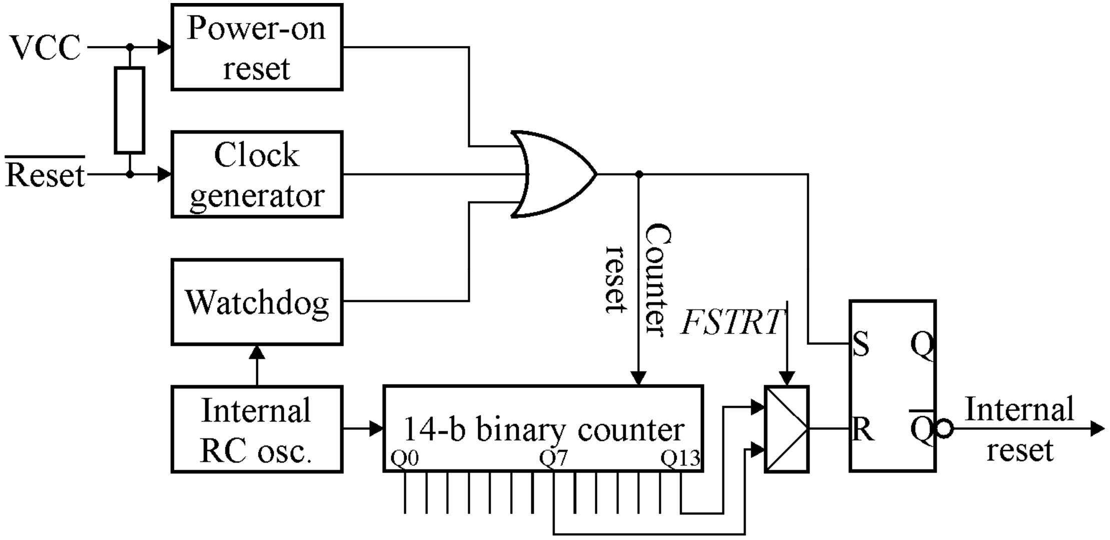
AVR (1)

- AVR – watchdog
  - WDTCR register
    - WDTOE – against accidental watchdog disable
      - WDTOE=1 -> WDE=0; WDTOE auto reset after 4 clk
    - WDE – watchdog enable (1)/disable (0)
    - $WDP_{0..2}$ – watchdog prescaler selection (16 µs÷2 ms)



AVR (1)

- AVR – reset
  - Reasons:
    - *Power-on Reset* – V_{CC} below V_{POT} level
    - *External Reset* – $\overline{\text{Reset}}$ input low ≥50 ns
    - *Watchdog Reset* – if watchdog not reset in time
  - Duration
    - FSTRT in FLASH
    - 16 ms/0.28 ms



  - May look different in other µc's, e.g. ATtiny

AVR (1)

- AVR registers
  - 32×8-bit universal registers
    - No outlined accumulator
      - Register-to-register transfer minimised
    - Almost full-freedom rule
      - Any register, any argument, any command
  - 3×16-b index registers
    - Made of register pairs
      - X: R27:R26; Y: R29:R28; Z: R31:R30
    - High-level language support
      - E.g., pointers to 2 arguments and a result
      - Postincrement/predecrement addressing modes
    - Some µc's have only 1 8-b index register (Z)
      - E.g. AT90S1200, some 8-pin µc's, etc.

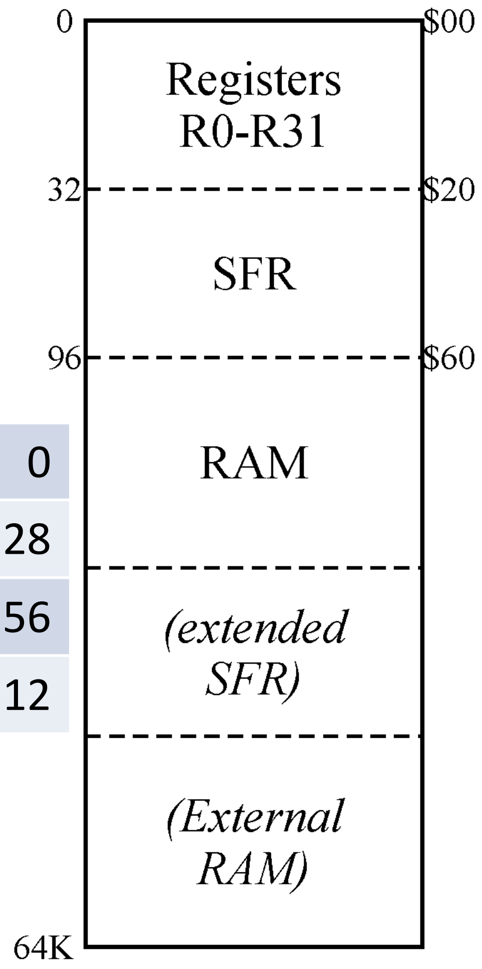
AVR (1)

- AVR registers
  - STATUS register
    - I – global interrupt enable (int.ack → 0; reti → 1)
    - T – intermediate bit during bit transfer
    - H – half carry (used in BCD)
    - S – sign (in U2); $\mathrm{S} = \mathrm{N} \oplus \mathrm{V}$
    - V – overflow (in U2)
    - N – negative (in U2)
    - Z – zero
    - C – carry

AVR (1)

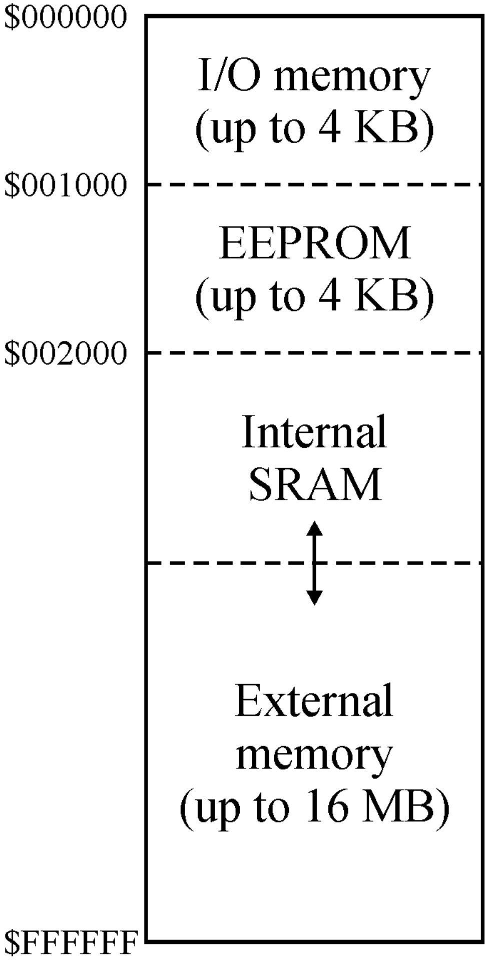
- AVR data memory
  - 32 registers
    - 3 index 16-b registers
  - SFR
    - Addressable as IO or memory
      - IO: $00÷$3F
      - Memory: $20÷$5F
  - RAM (if exists)
  - Extended SFR
    - If not enough space in SFR area
  - Extended RAM
    - If such can be attached

AT90S1200	0
AT90S2313	128
AT90S4414	256
AT90S8515	512



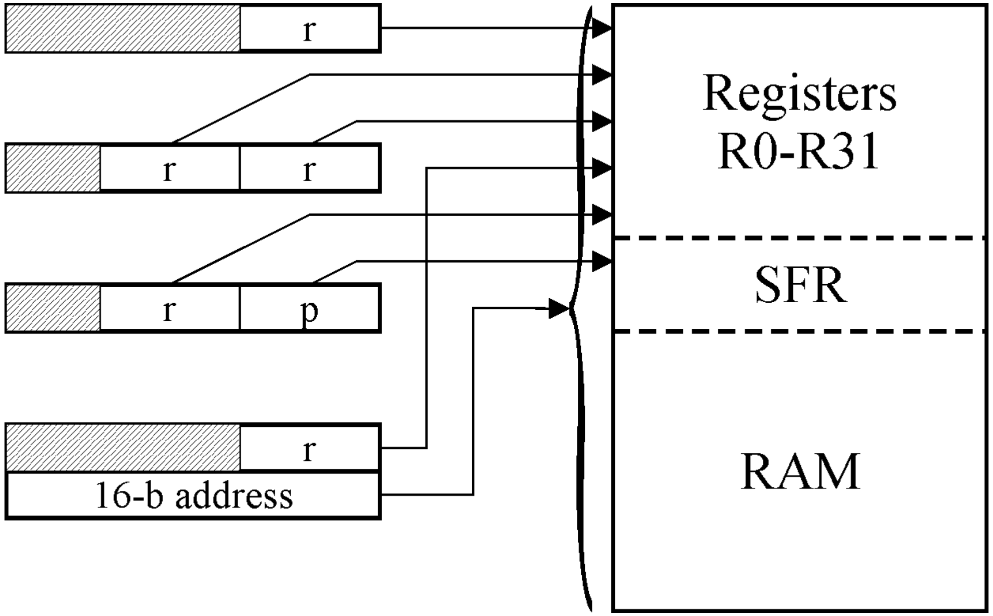
AVR (1)

- AVR data memory in xmega family
  - 24-b address range (16 MB space)
  - 32 registers
    - 3 index 16-b registers
  - Additional registers
    - RAMPX, RAMPY, RAMPZ – bits 16..23 for X, Y, Z registers
      - RAMP = RAM page?
    - RAMPD – direct address extension
    - EIND – extended indirect (jump, calls) for Z register



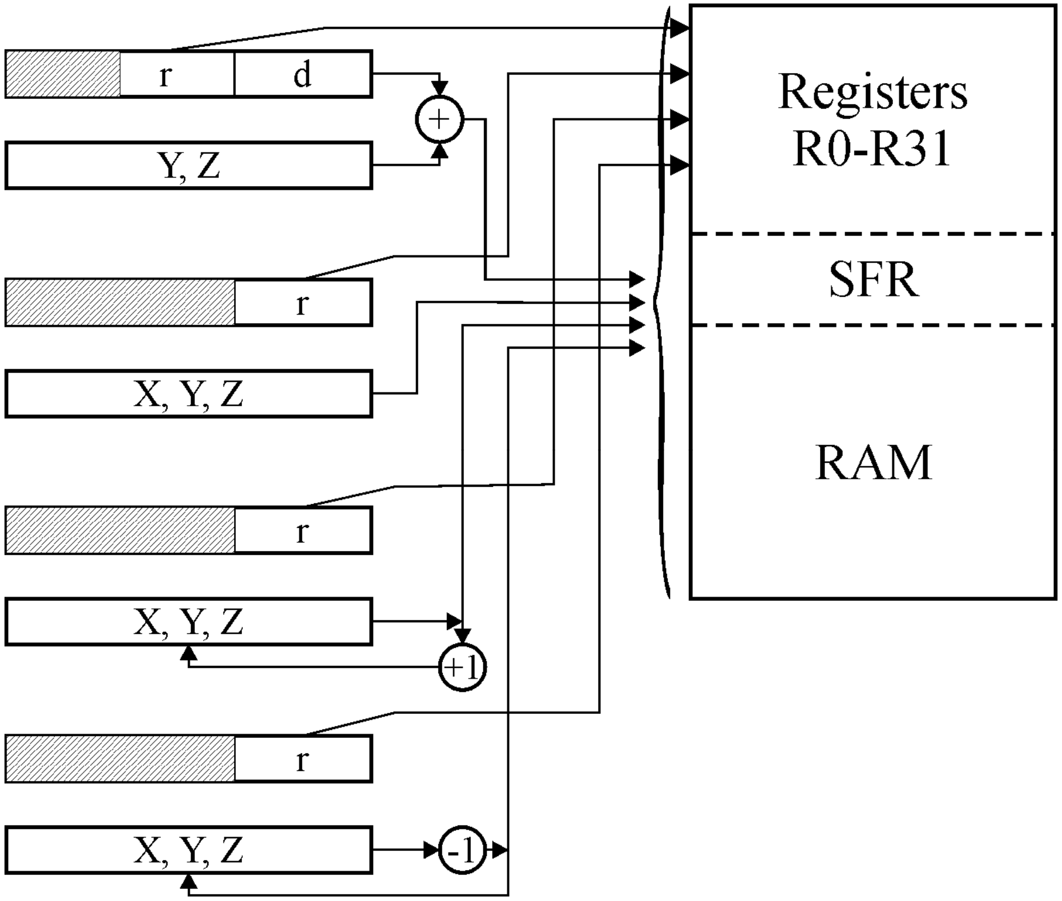
AVR (1)

- AVR data memory
  - Direct addressing modes
    - Register direct
      - `inc R0`
    - 2 Registers direct
      - `or R1, R2`
    - IO direct
      - `out PortB, R3`
    - Data direct
      - `lds R4, $65`



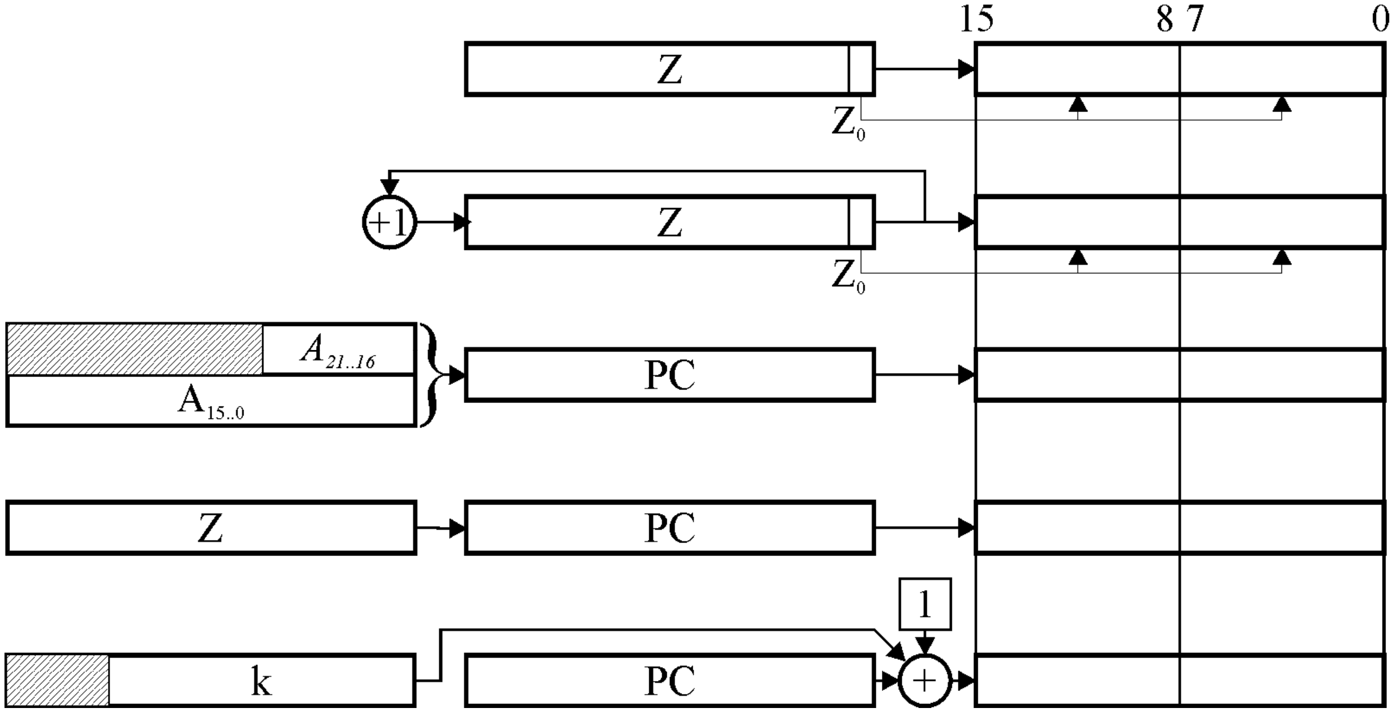
AVR (1)

- AVR data memory
  - Indirect addressing modes
    - Indirect with displacement
      - `ldd R6, Y+5`
    - Indirect
      - `st Y, R7`
    - Indirect with postincement.
      - `st Y+, R7`
    - Indirect with predecrement.
      - `st -Y, R7`



AVR (1)

- AVR program memory
  - Addressing modes
    - Constant (`lpm`); ROM[Z]→R0
    - Indirect program (`ijmp`, `icall`); goto [Z]
    - Relative program (`rjmp`, `rcall`)



AVR (1)

- AVR – stack
  - Internal data memory exists
    - Stack mapped into data memory
    - SP must be initialised at program start
    - Software available, `push`/`pop` implemented
  - Internal data not implemented (RAM=0)
    - Hardware stack as a separate memory
      - E.g., 90S1200: 3 cells×9 bits
    - No software access, no `push`/`pop` commands
    - Stack overflow not controlled/signalled
    - Only return address for interrupts or calls

AVR (1)

- AVR – EEPROM
  - Registers:
    - EECR control register
      - EEMWE – 1→master write enable
        - » Automatically reset after 4 Tclk
      - EEWE – 1→ write start
      - EERE – 1→ read start
    - EEDR – data register
    - EEAR – address register
  - !!! EEAR=0 after reset
    - 00h address accidentaly written
      - Disable reset during EEPROM write
      - Don’t use 00h address in EEPROM

AVR (1)

- AVR – I/O ports
  - Registers:
    - DD – data direction (rd/wr)
    - Port – register (rd/wr)
    - Pin – pin state (read-only, not a real register)
  - Port configuration

DD	Port	Direction	Pull-up	Comment
0	0	Input	No	Hi-impedance input
0	1	Input	Yes	Input/current source if pulled low outside
1	0	Output	No	Push-pull output – 0 state
1	1	Output	No	Push-pull output – 1 state

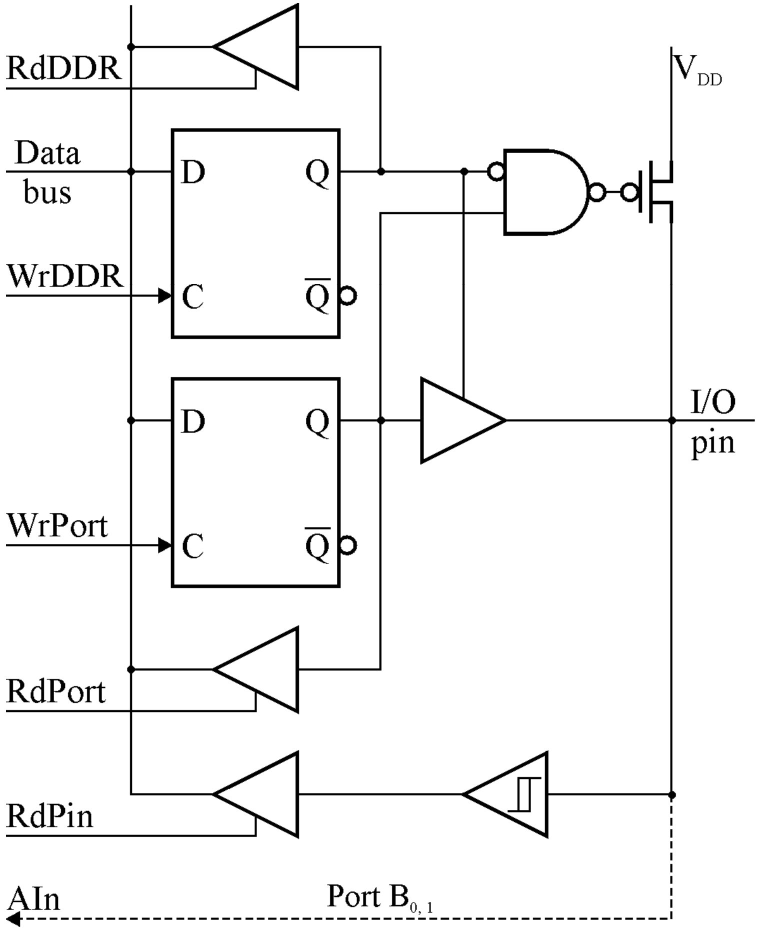
AVR (1)

- AVR – I/O ports
  - PortB
    - 8-b, 20 mA output current
    - Alternative functions
      - PB0: AIN0 – analogue comparator positive input
      - PB1: AIN1 – analogue comparator negative input
      - PB3: OC1 – output compare (match output) for TC1
      - PB5: MOSI – SPI input
      - PB6: MISO – SPI output
      - PB7: SCK – SPI clock

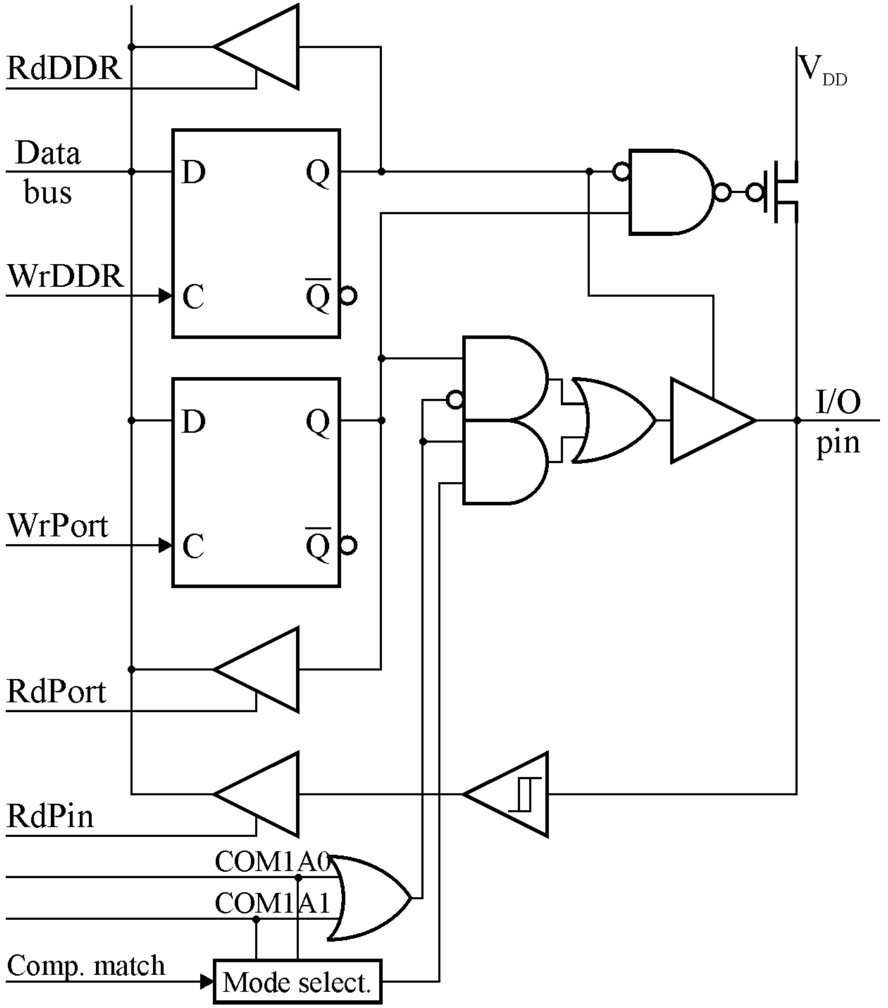
      } for serial programming

AVR (1)

- AVR – example I/O ports structure



Port $B_{0, 1, 2, 4}$



Port B_3

AVR (1)

- AVR – I/O ports
  - PortD
    - 8-b, 20 mA output current
    - Alternative functions
      - PD0: RxD – UART input
      - PD1: TxD – UART output
      - PC2: Int0 – external interrupt
      - PD3: Int1 – external interrupt
      - PD5: T0 – TC0 clock input
      - PD6: T1 – TC1 clock input
      - PD7: ICP – TC1 capture trigger input
    - Structure similar to PB5

AVR (1)

- AVR – I/O ports
  - PortA (e.g., AT90S4414, AT90S8515)
    - Alternative functions
      - Address low/data bus multiplexed
      - More complex structure than PB5
  - PortC (e.g., AT90S4414, AT90S8515)
    - Alternative functions
      - Address high bus
      - A little more complex structure than PB5