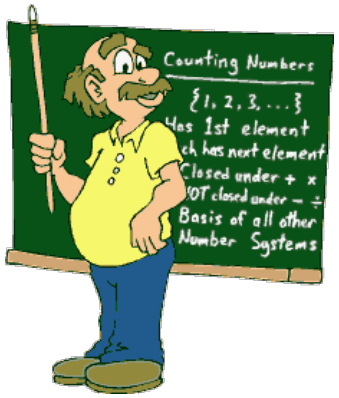
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

MSYS

Microcontroller Systems

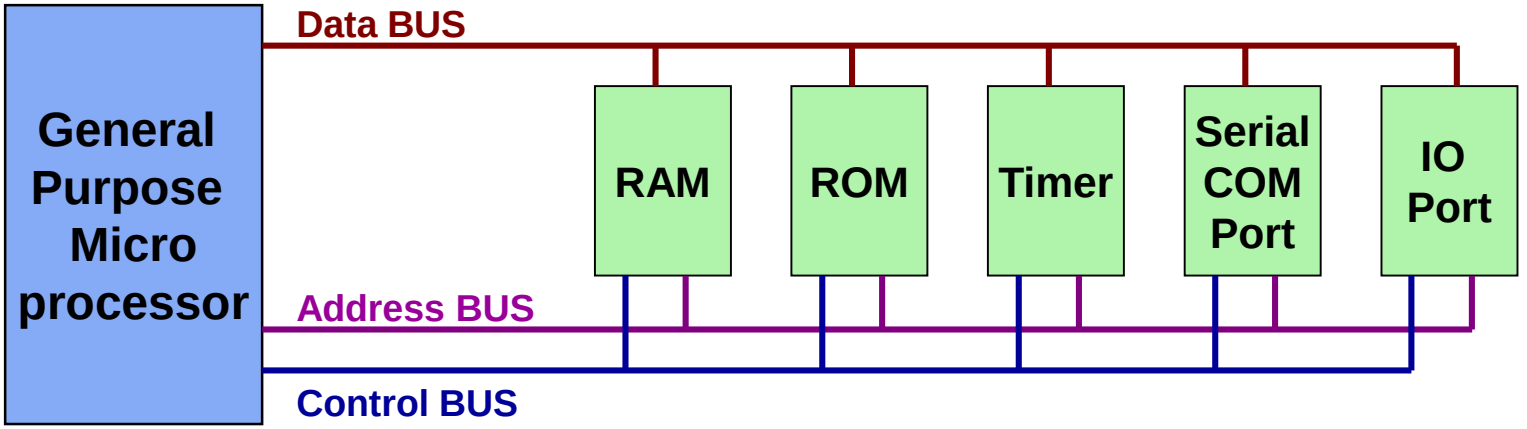
Lektion 4: AVR arkitektur



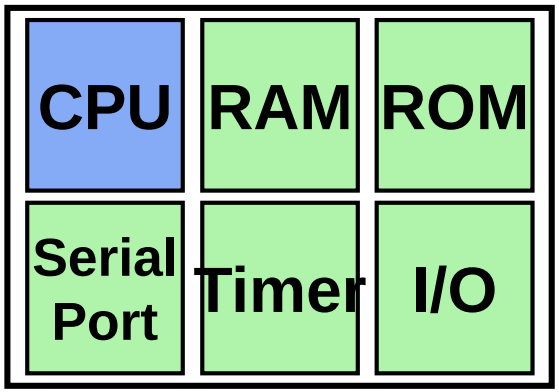
Version: 4-9-2017, Henning Hargaard

General Purpose Microprocessors vs. Microcontrollers

- General Purpose Microprocessors



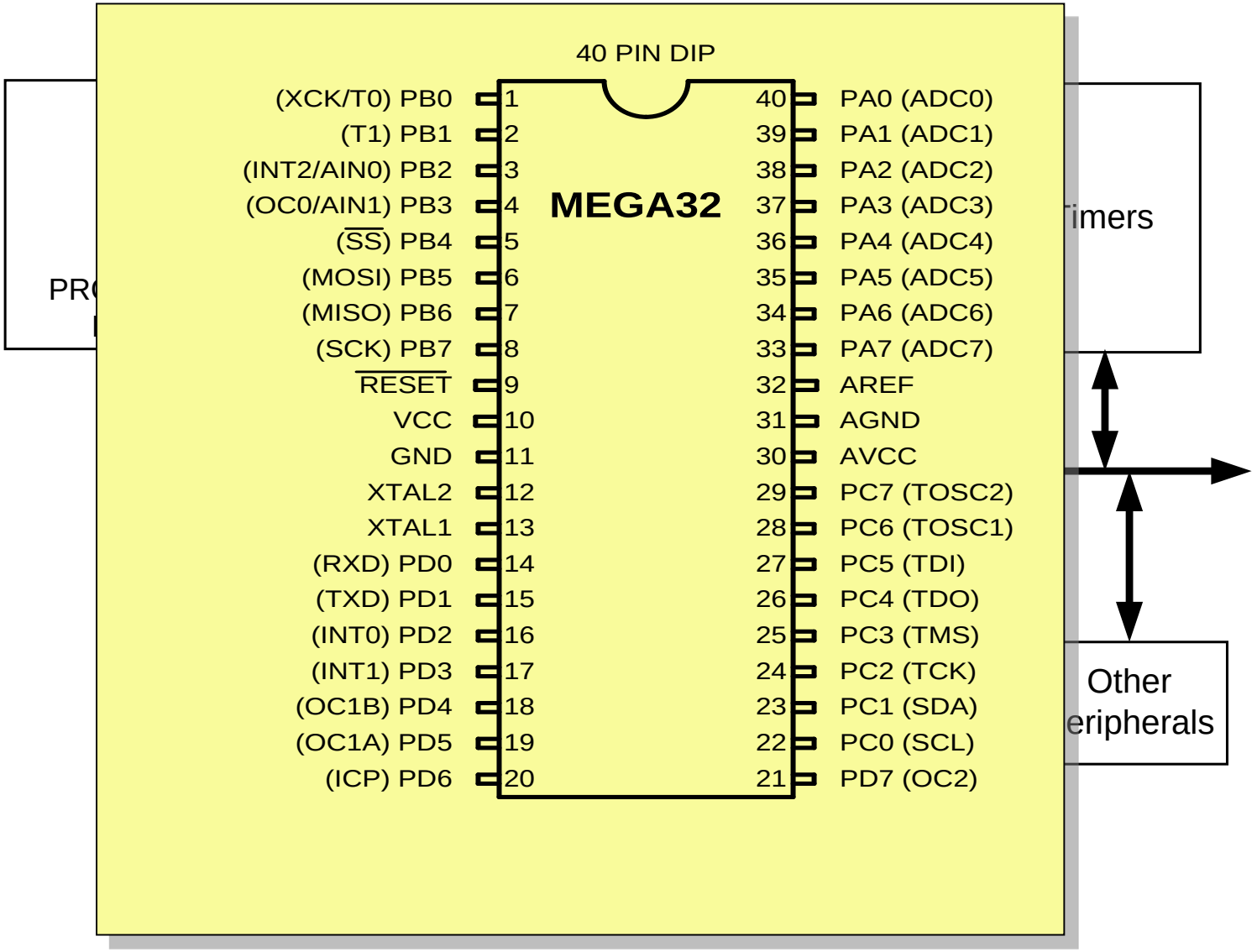
- Microcontrollers



Common microcontrollers

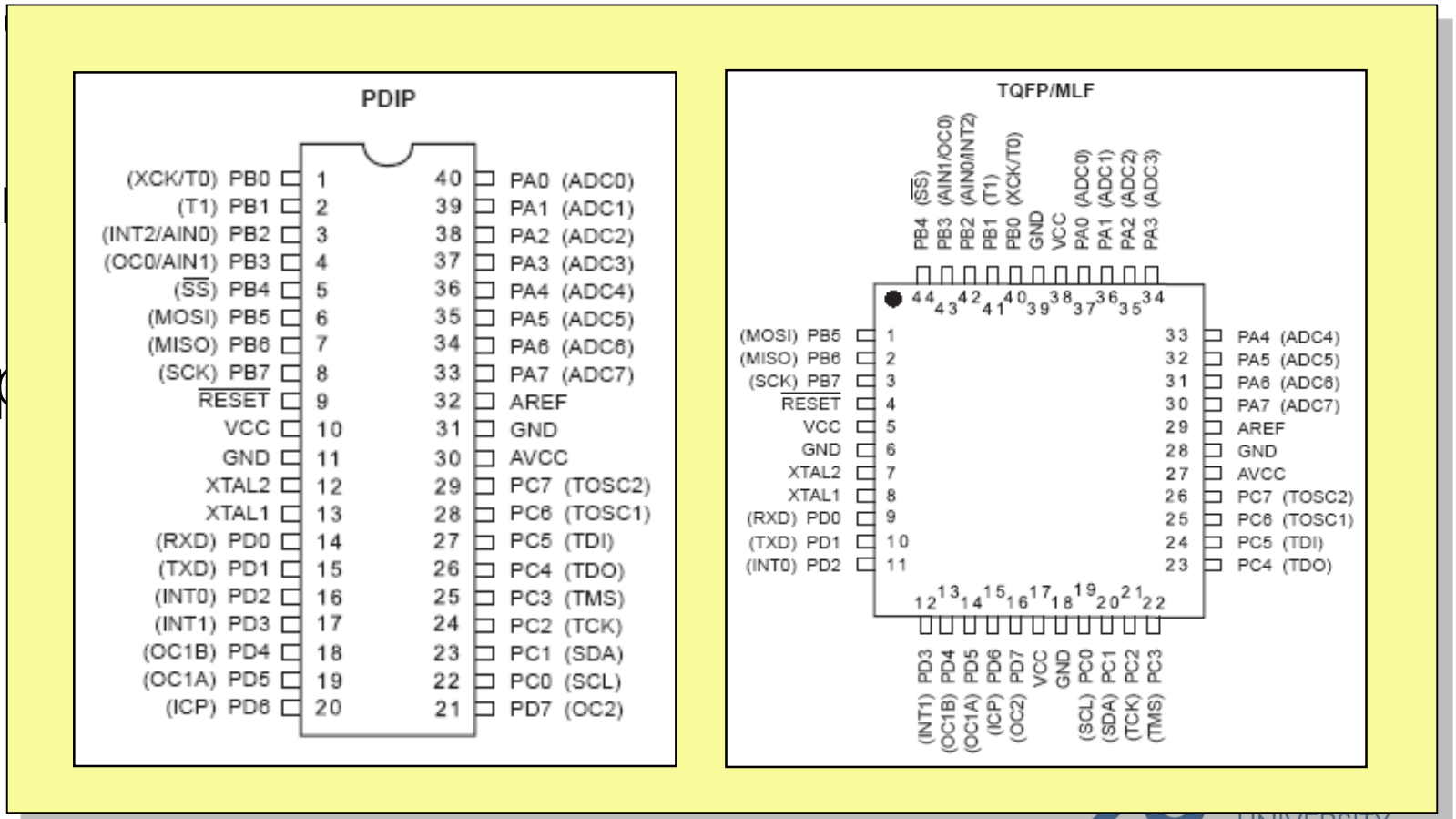
- 8-bit microcontrollers
  - AVR
  - PIC
  - HCS12
  - 8051
- 32-bit microcontrollers
  - ARM
  - PIC32

AVR internal architecture

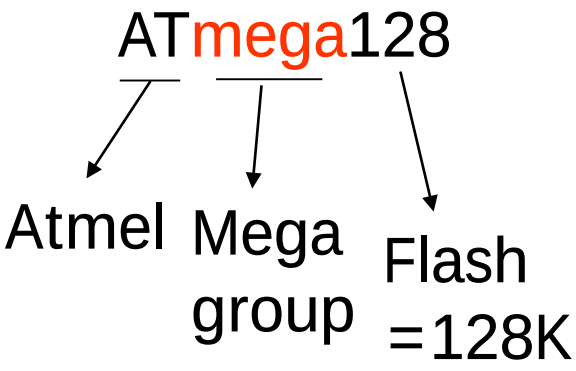


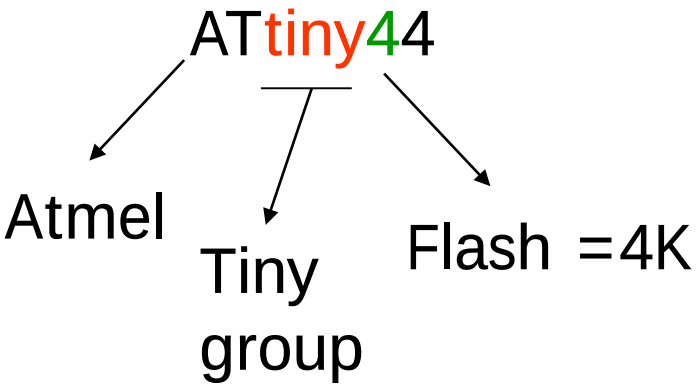
AVR different groups

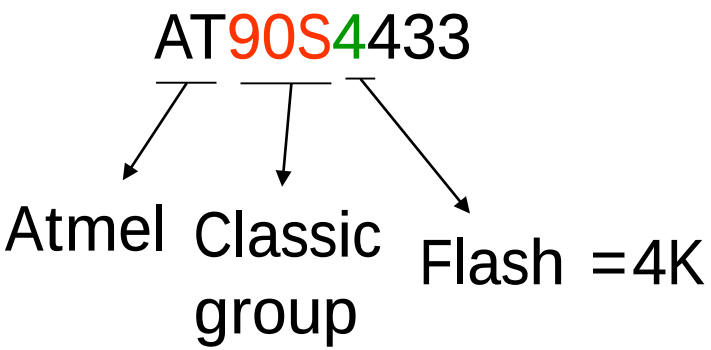
- Classic AVR
  - e.g. AT90S2313, AT90S4433
- M
  -
- Ti
  -
- Sp
  -



The AVR part numbers

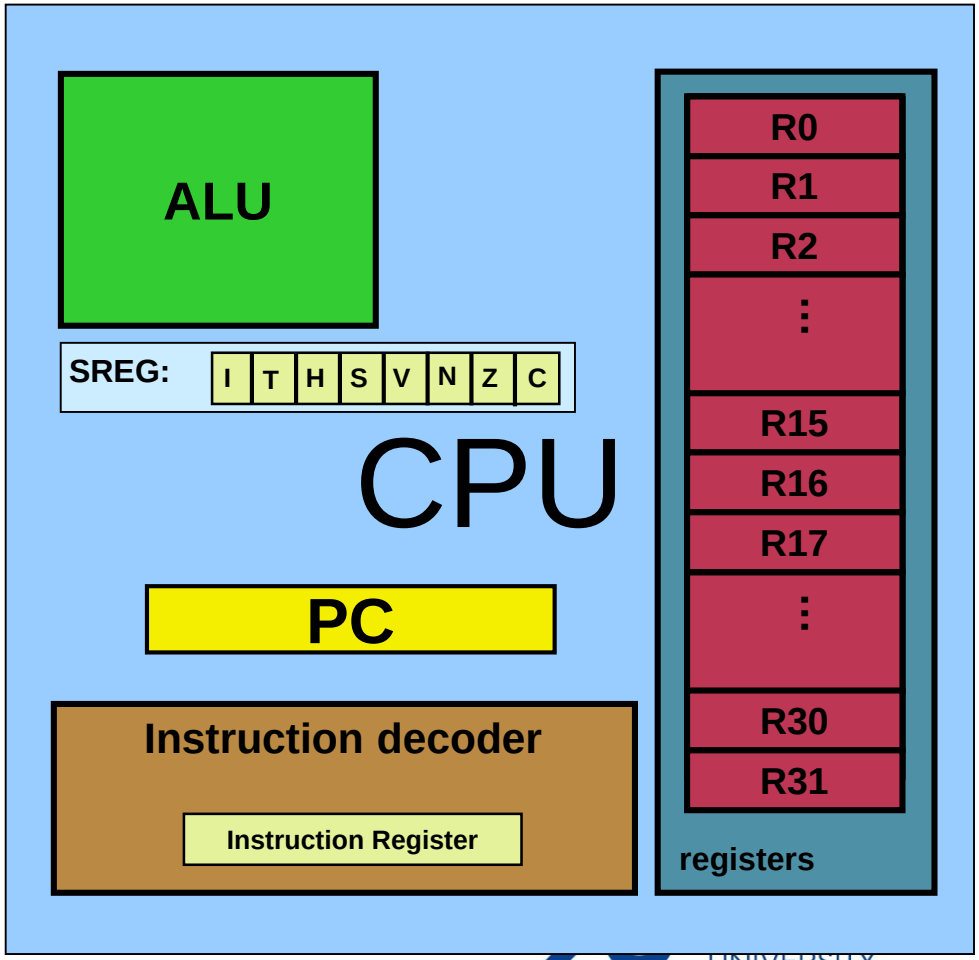






AVR's CPU

- AVR's CPU
  - ALU
  - 32 General Purpose registers (R0 to R31)
  - PC register
  - Instruction decoder

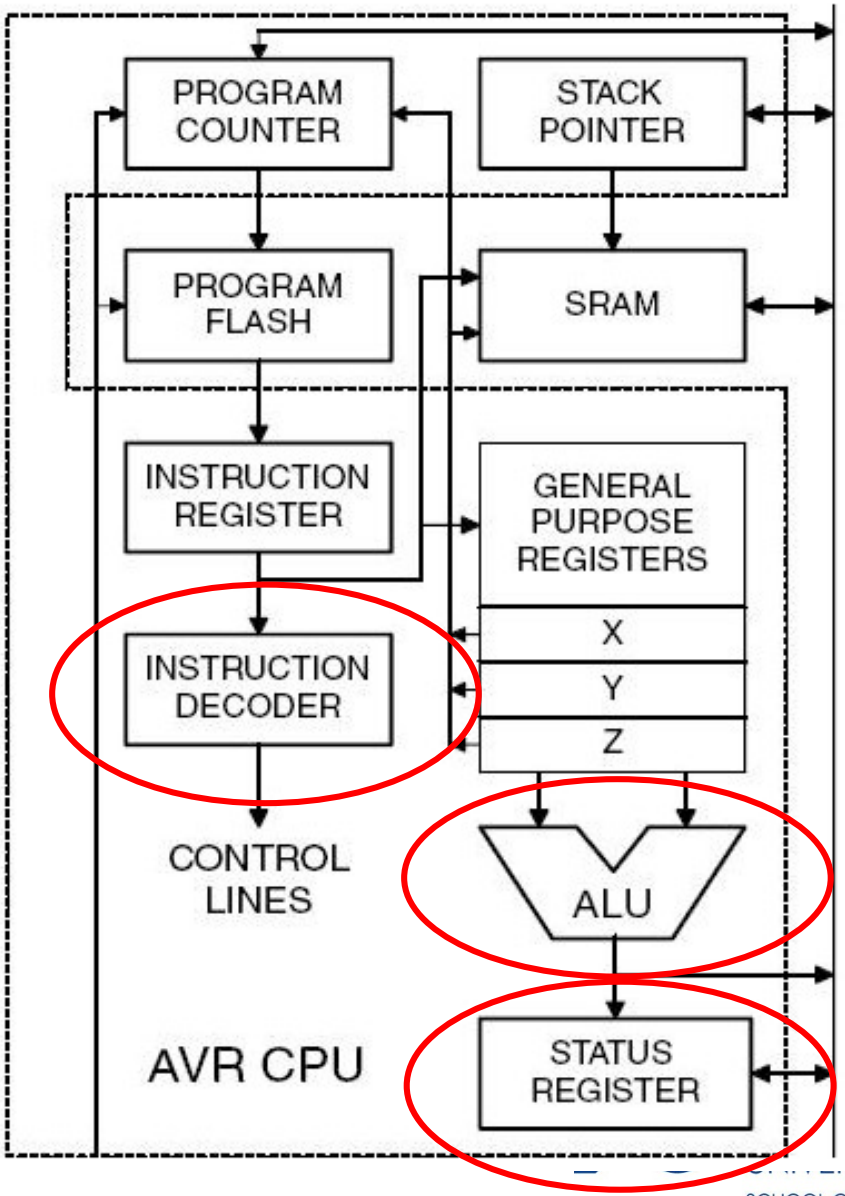


AVR CPU og maskin - kode

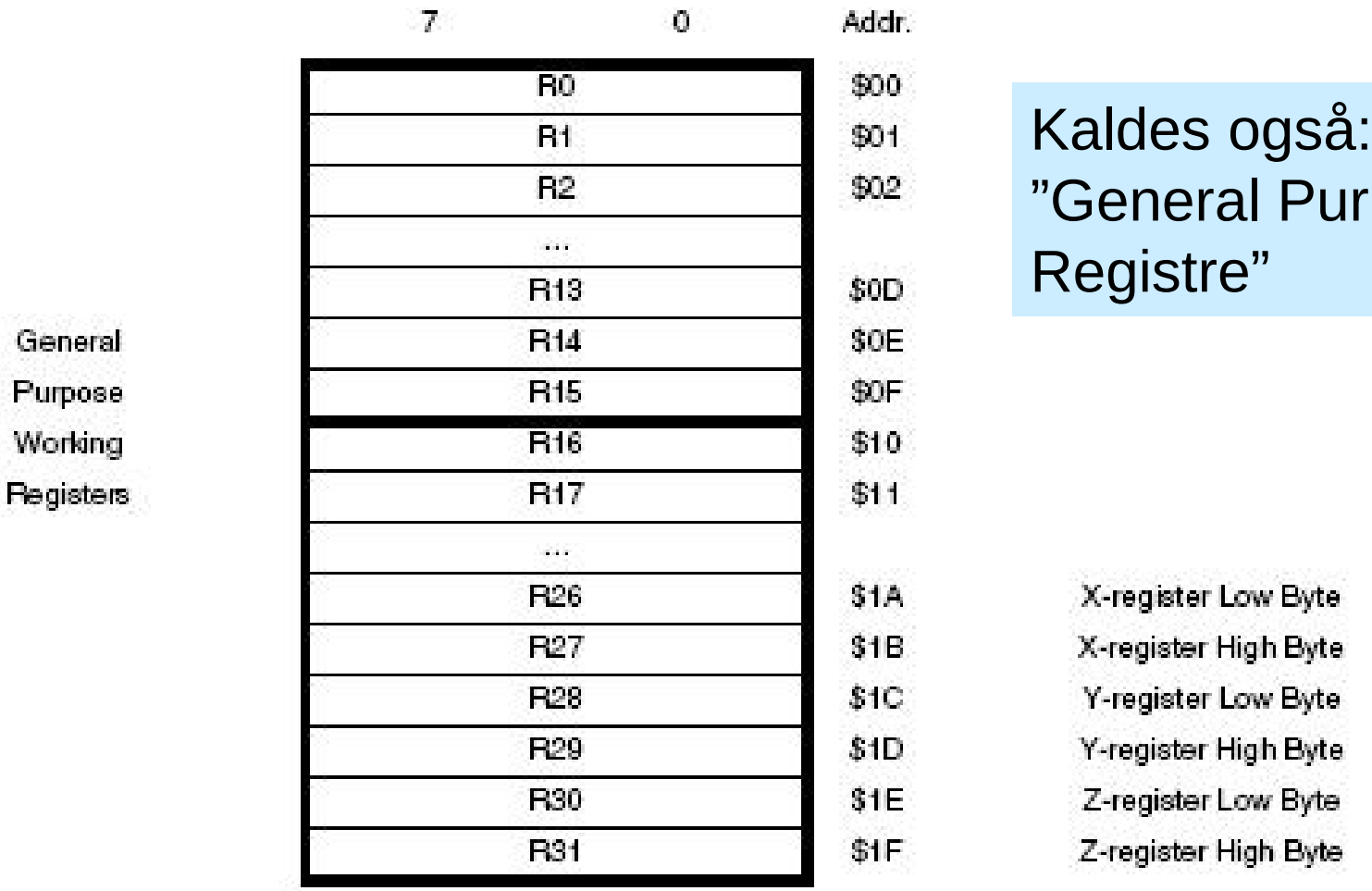
Instruktioner er koder bestående af 0'er og 1'taller !

ALU foretager beregninger.

Status register ændres ved nogle beregninger.

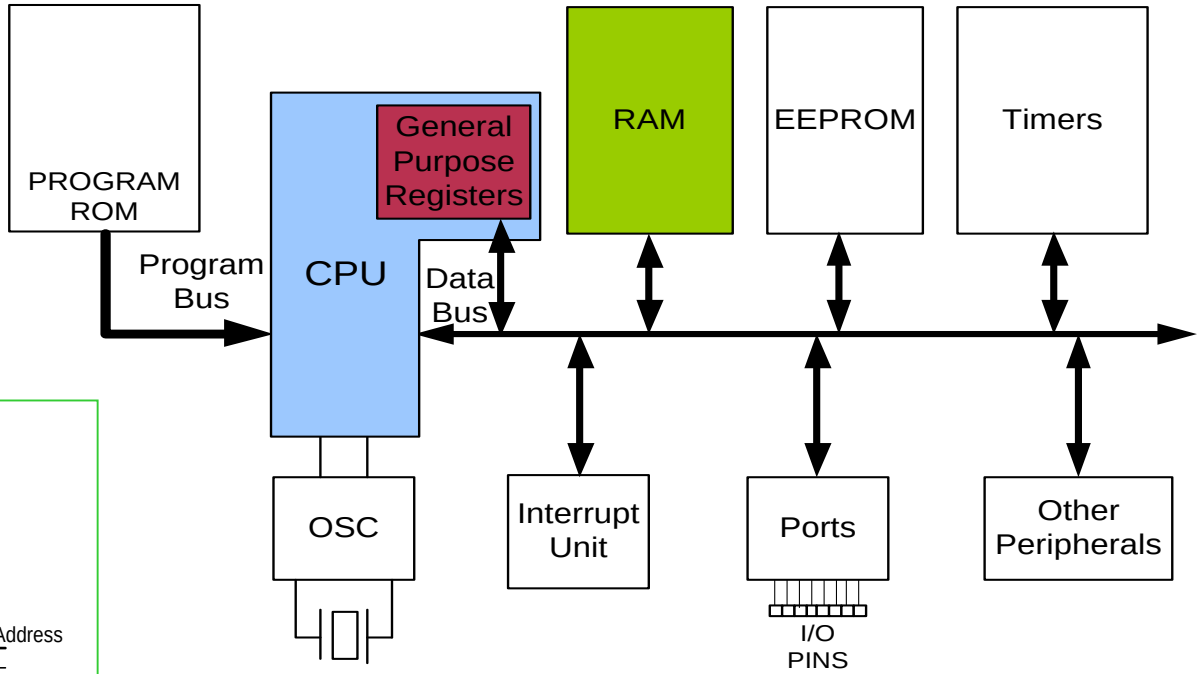


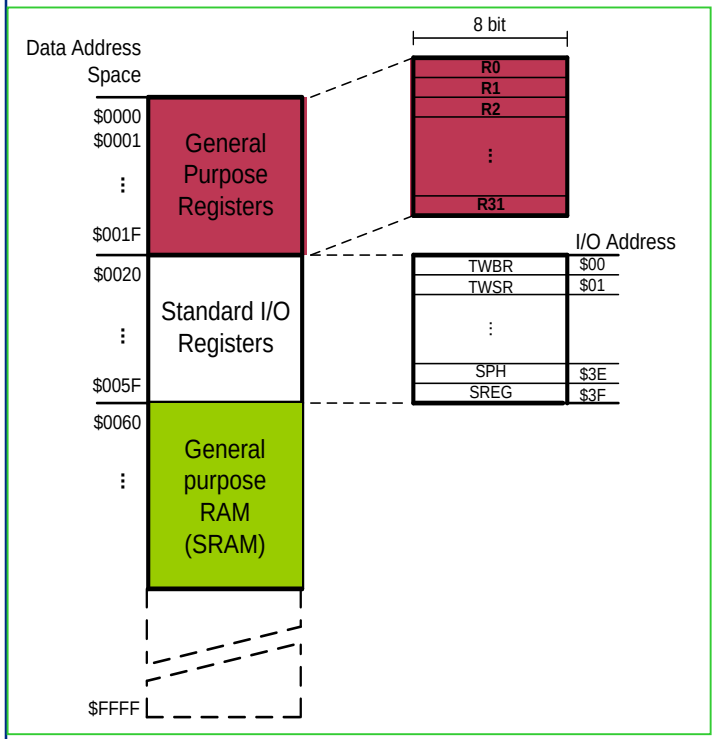
AVR arbejds-registre



Kaldes også: ”General Purpose Registre”

AVR: Data Spaces

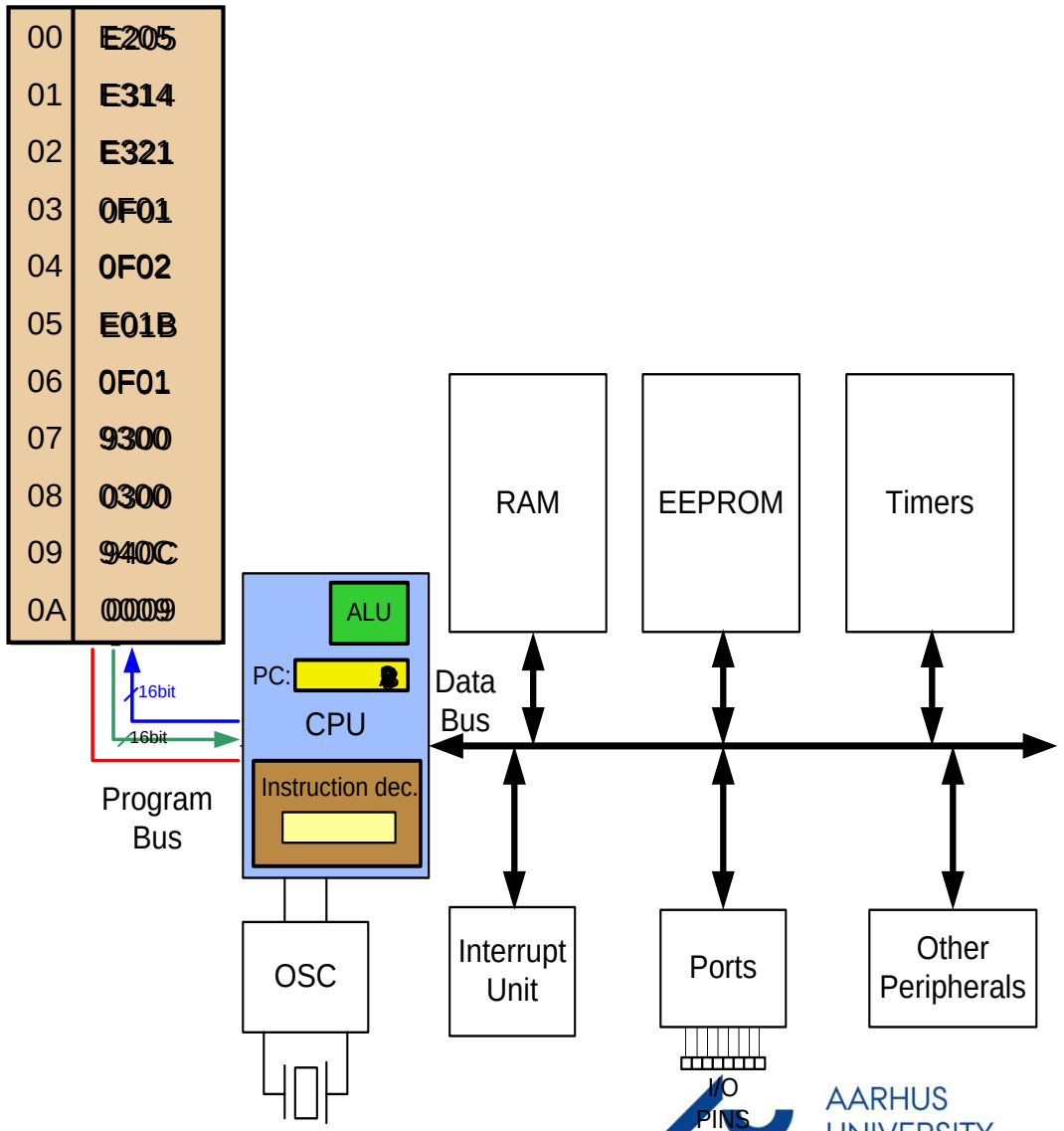




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

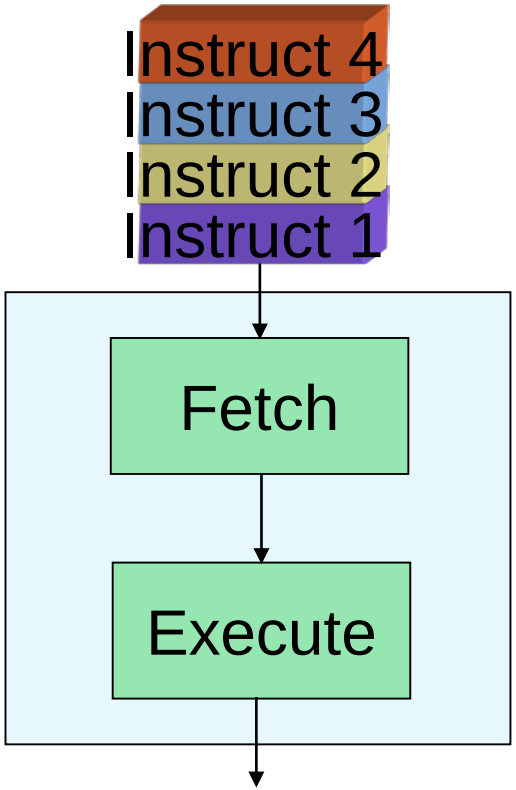
Flash memory and PC register

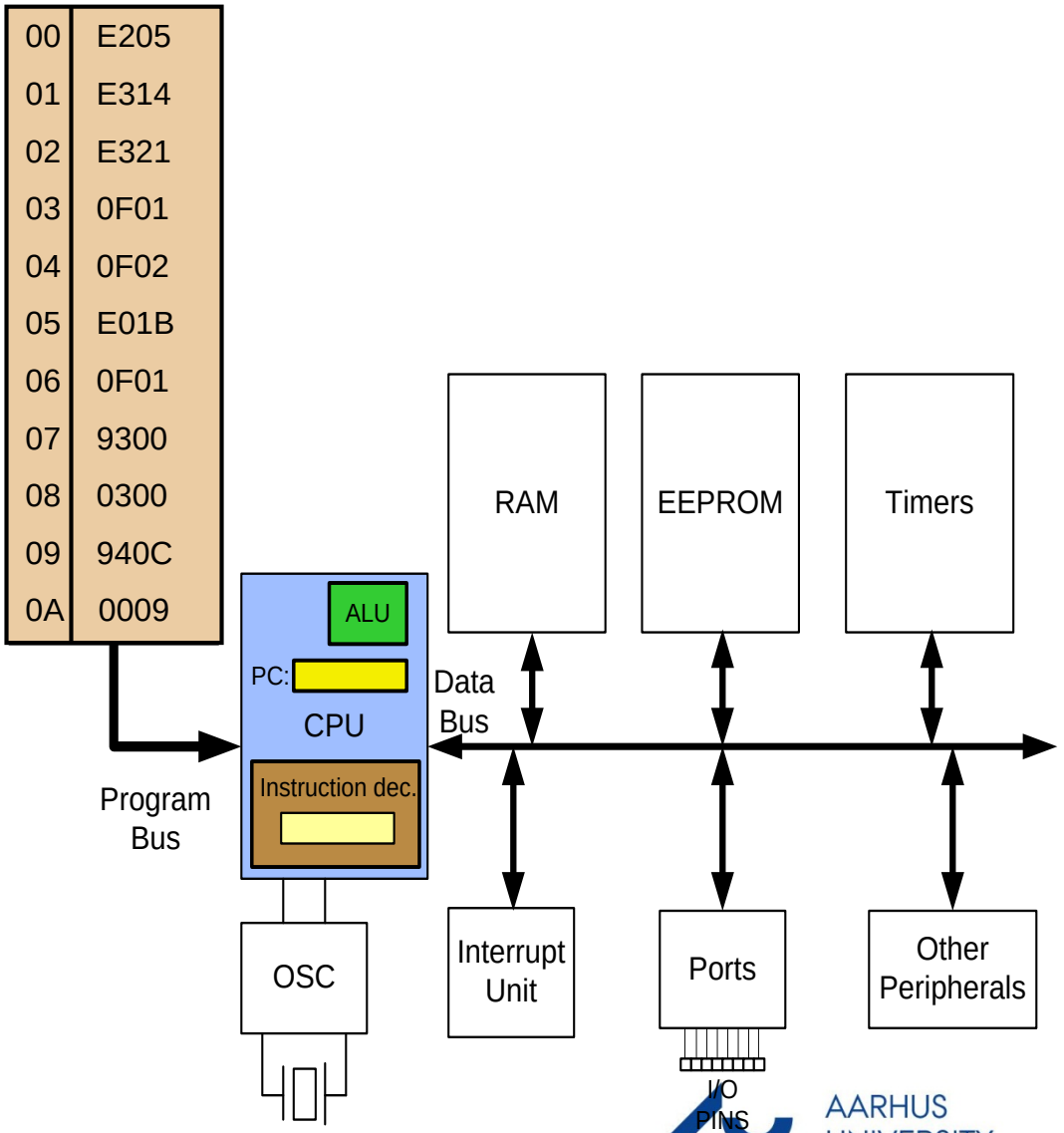
```
      LDI   R16, 0x25
      LDI   R17, $34
      LDI   R18, 0x31
      ADD   R16, R17
      ADD   R16, R18
      LDI   R17, 11
      ADD   R16, R17
      STS   SUM, R16
HERE: JMP  HERE
```



Fetch and execute

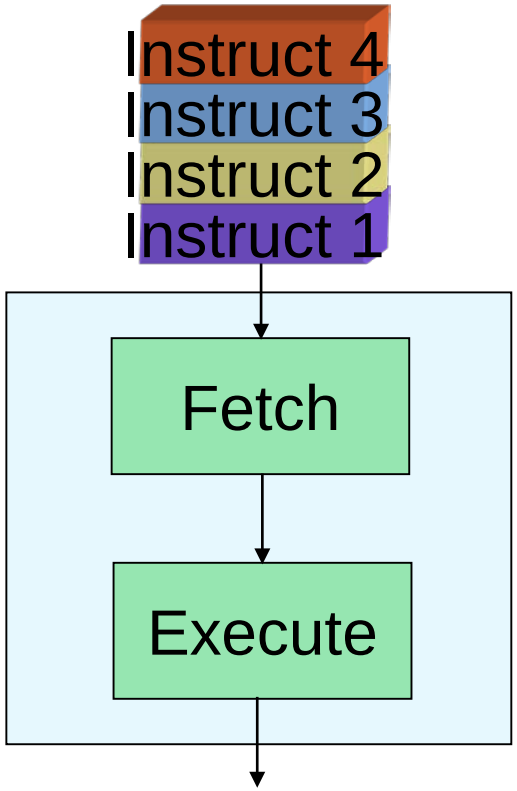
- Old Architectures

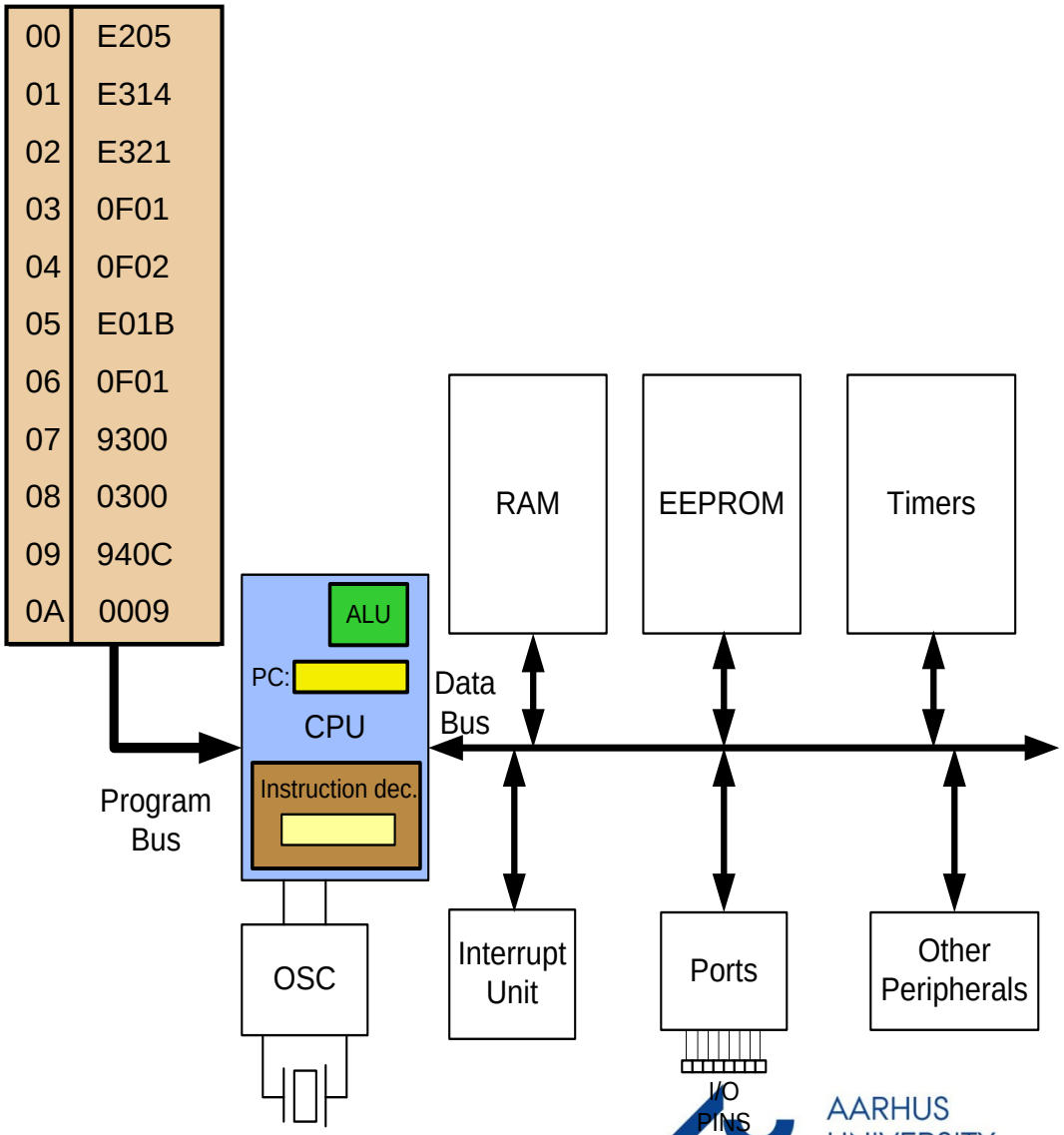




Pipelining

- Pipelining



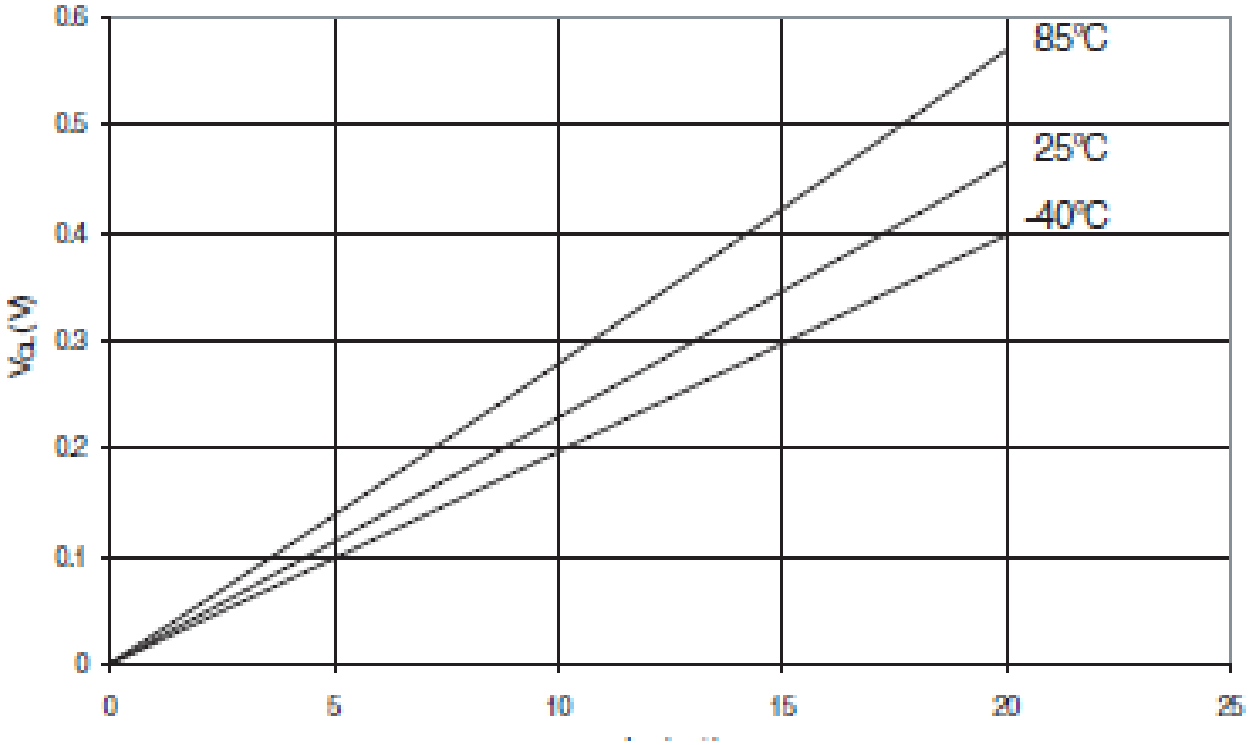


How to speed up the CPU

- Increase the clock frequency
  - More frequency ➔ More power consumption & more heat
  - Limitations
- Change the architecture
  - Pipelining
  - RISC

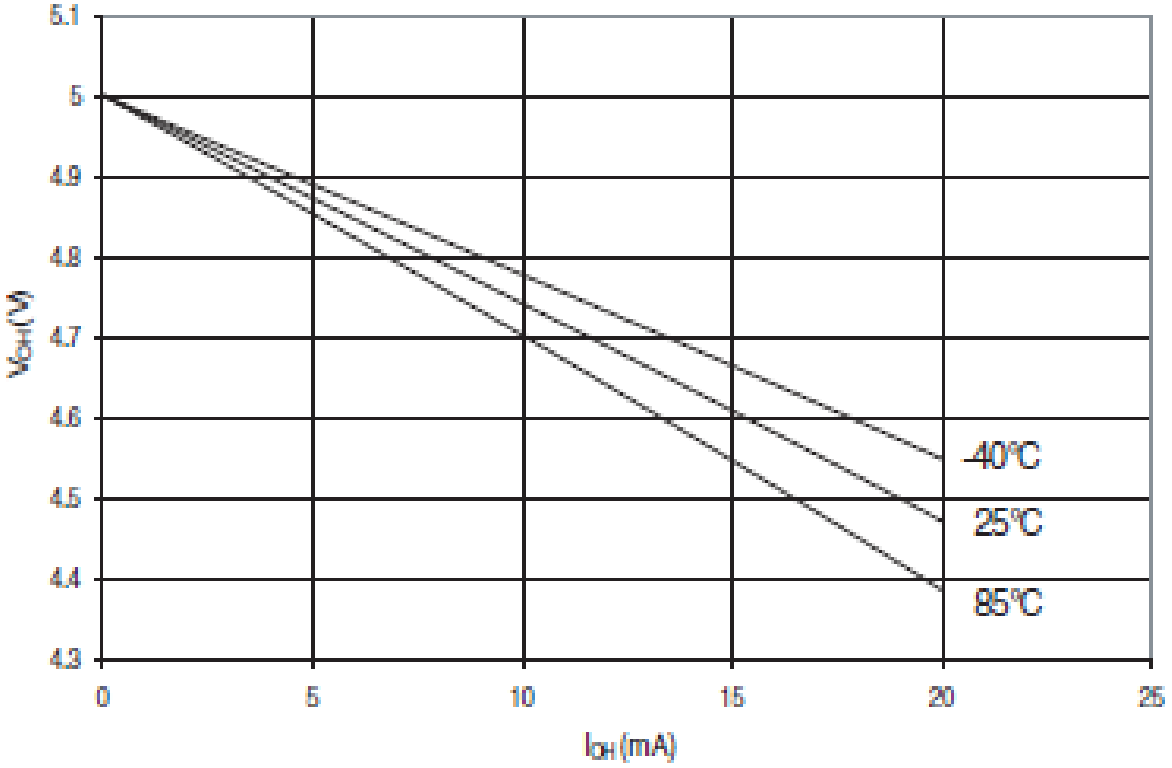
Low er ikke (helt) 0 volt

Figure 32-23. I/O Pin Output Voltage vs. Sink Current (V_{CC} = 5V)



High er ikke (helt) 5 volt

Figure 32-25. I/O Pin Output Voltage vs. Source Current (V_{CC} = 5V)



Programmet fra LAB1

```
;********** MSYS, LAB1 **********
;******* Henning Hargaard *******
;*******  14.august 2015  *******
;********************************

;********** INITIERING **********
   LDI  R16,HIGH(RAMEND) ;Initialize Stack Pointer
   OUT  SPH,R16
   LDI  R16,LOW(RAMEND)
   OUT  SPL,R16
   SER  R16              ;PORTB = Outputs
   OUT  DDRB,R16

;********** PROGRAM-LOOP ********
   CLR  R16
LOOP:
   LDI  R17,9            ;R17 = 9
   ADD  R16,R17          ;R16 = R16 + R17
   CALL DISP_AND_DELAY   ;Display R16
   JMP  LOOP             ;Jump to "LOOP"


   ..........
```

```
..........

;********** DISPLAY R16 *********
;********** AND DELAY ***********
DISP_AND_DELAY:
   MOV  R17,R16
   OUT  PORTB,R17
   CLR  R17
   CLR  R18
   LDI  R19,100
AGAIN:
   DEC  R17
   BRNE AGAIN
   DEC  R18
   BRNE AGAIN
   DEC  R19
   BRNE AGAIN
   RET
;********************************
```

Slut på MSYS lektion 4