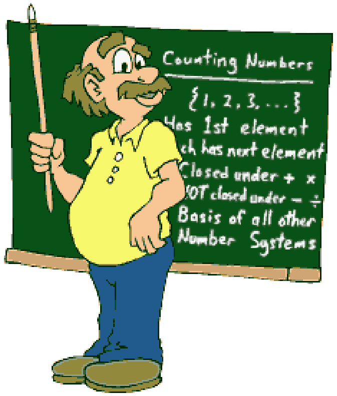
AARHUS
UNIVERSITY

SCHOOL OF ENGINEERING

MSYS

Microcontroller Systems

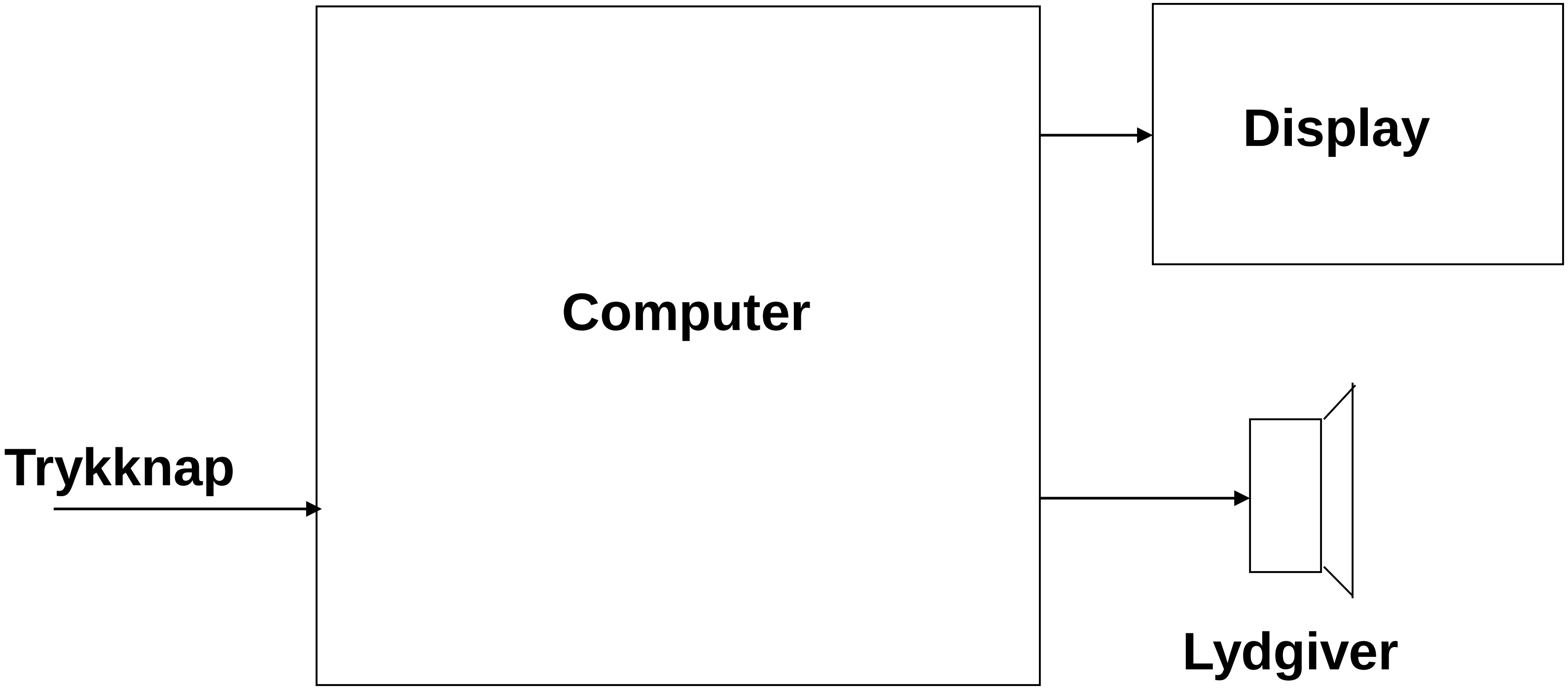
Lektion 16: Interrupts



Version: 31-10-2017, Henning Hargaard

Interrupt = (Midlertidig) afbrydelse

Eksempel



* Computeren beregner hele tiden ”noget”, og viser det på displayet.

* Hvis man trykker på trykknappen, skal lydgiveren sige ”bib”.

Løsning uden brug af interrupt

```
int main()
{
  [Klargør al hardware]
  while(1)
  {
    [Beregn noget]
    [Vis det på displayet]
    if ([Der er trykket på knappen]
      [Sig bib med lydgiveren]
  }
}
```

Hvad er ulempen?

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Hvad er interrupts ?

- Et interrupt er en midlertidig afbrydelse af den igangværende programafvikling.
- Typisk startes et interrupt via et hardware-signal (en hændelse).
- Under interrupt'et afvikles en speciel funktion, der kaldes en interrupt-rutine (eller interrupt service rutine = **ISR**). Denne er skrevet af programmøren.
- Efter interrupt'et vendes tilbage til den afbrudte programafvikling.

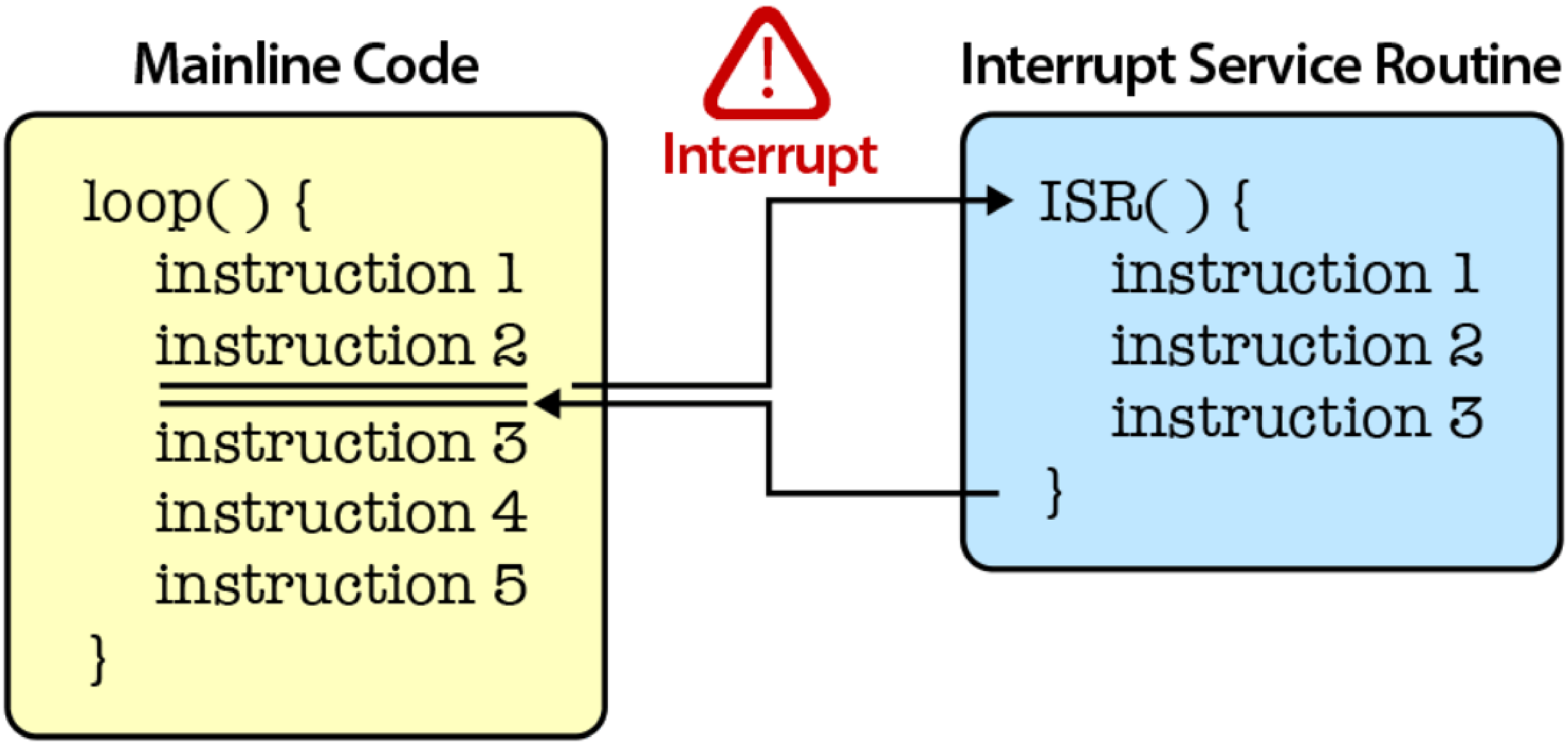
Løsning med brug af interrupt

```
// Denne funktion kaldes AUTOMATISK af HARDWARE
// hver gang, der trykkes på knappen
ISR (INT0_vect)
{
  [Sig bib med lydgiveren]
}


int main()
{
  [Klargør al hardware]
  [Klargør (enable) interrupt]
  while(1)
  {
    [Beregn noget]
    [Vis det på displayet]
  }
}
```

Fordele / ulemper?

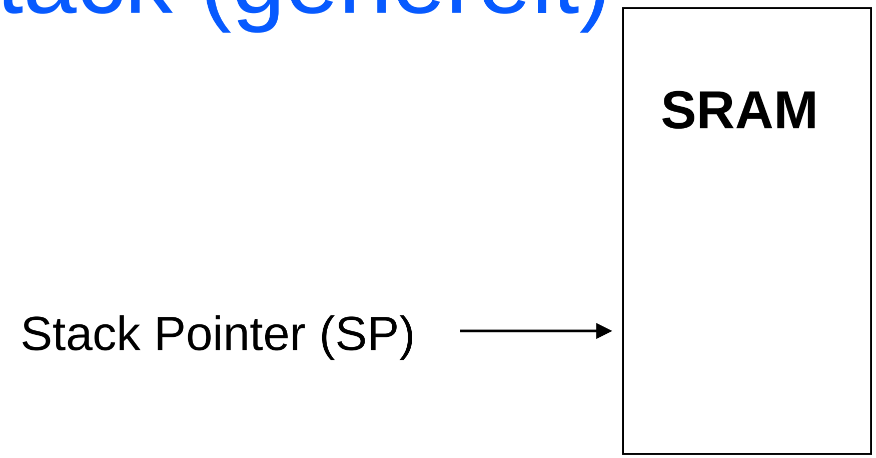
Interrupt - sekvensen



Interrupt - sekvensen

1. Igangværende instruktion færdiggøres.
2. Programtælleren gemmes på stacken.
   (varer 4 clk-cykles)
3. Jump til interrupt-rutinen.
   (varer 3 clock-perioder)
4. Global interrupt enable flag nulstilles.
   Hvis ”sei” i interrupt-rutinen => mulighed for ”nested interrupts”.
5. RETI => Programtælleren hentes fra stacken og global interrupt enable flag sættes igen.
   (varer ialt 4 clk-perioder).

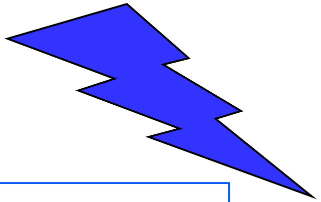
Hardware Stack (generelt)


SRAM

Stack Pointer (SP)

- Stack pointer’en er et vigtigt register, som holder styr på, hvor ”toppen af stacken” er (d.v.s. næste ledige plads (adresse)).
- Hver gang data skrives til stacken, flyttes SP automatisk **tilbage** (dekrementeres).
- Hver gang data hentes fra stacken, flyttes SP automatisk **frem** (inkrementeres).

Interrupts og stacken



Ved HW-interrupt gemmes adressen på næste instruktion **automatisk** på stacken !
SP **de**krementeres.

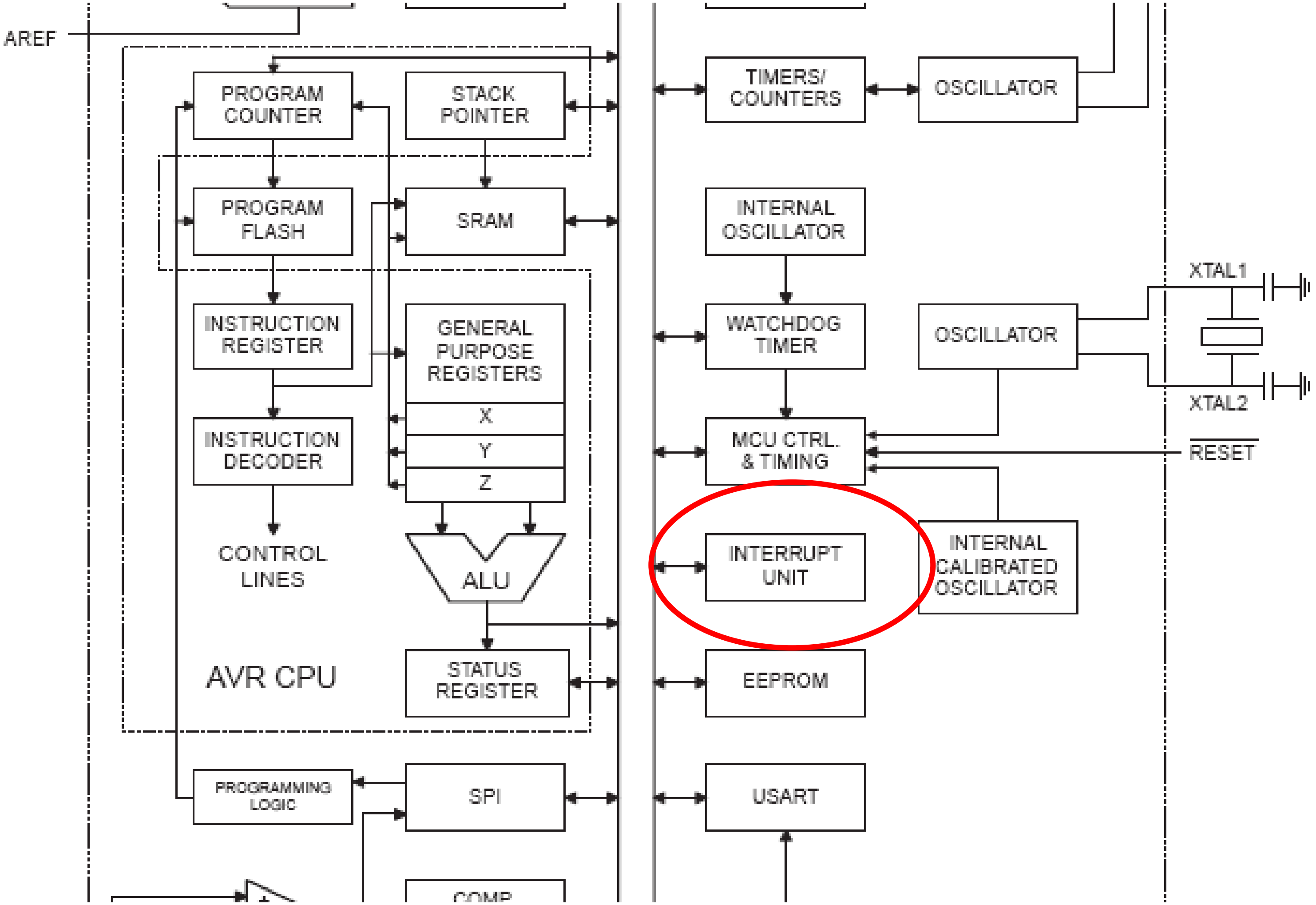
```
Interrupt:
          MOV R0,R4
          INC R7
          MOV R1,R3
          RETI
```

Programtælleren (PC) loades fra stacken.
SP **in**krementeres.

- Ved interrupts anvendes stacken automatisk, så der fortsættes i programmet, hvor man blev afbrudt.

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

AVR interrupt unit



AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Mega32: Interrupt vektorer

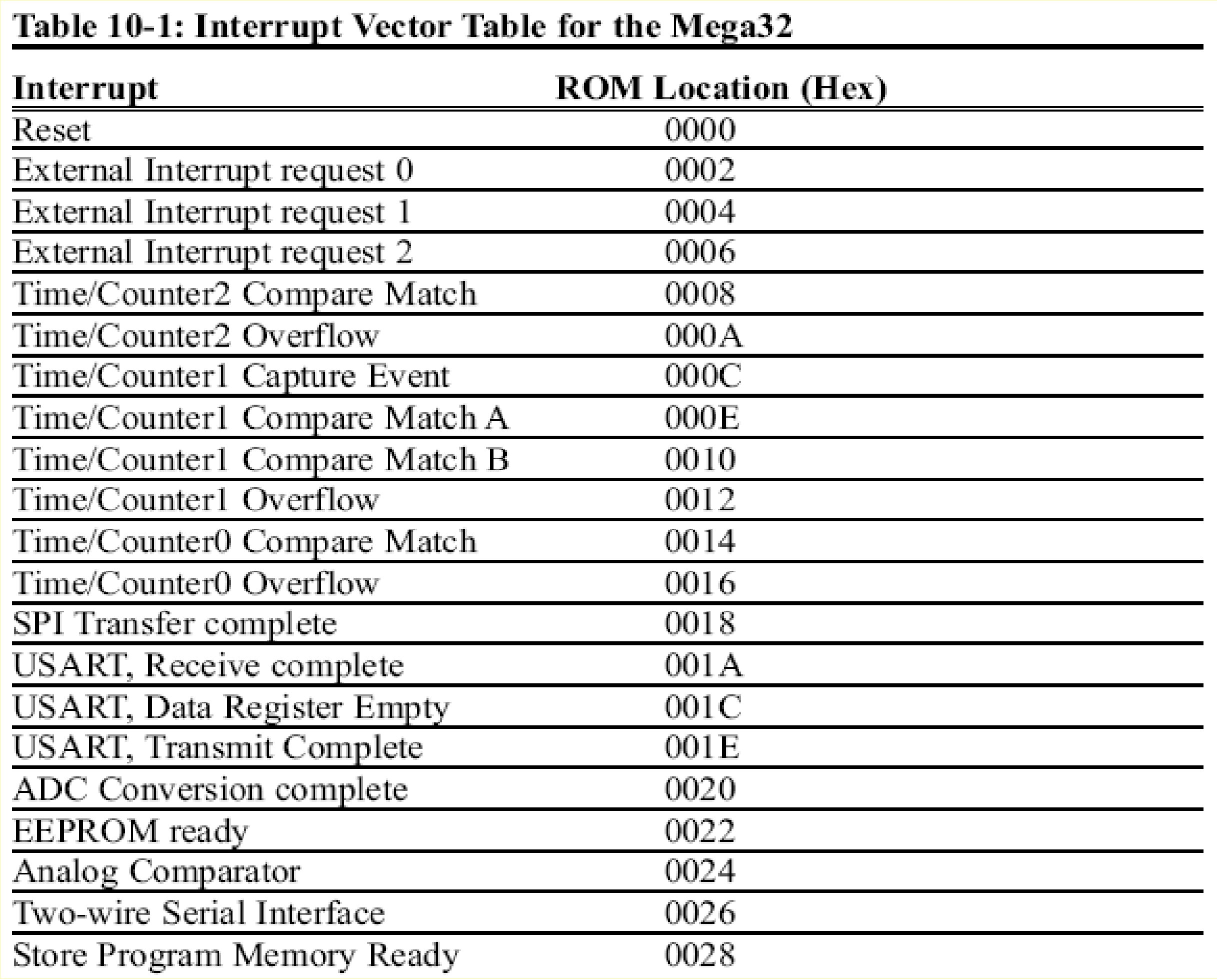
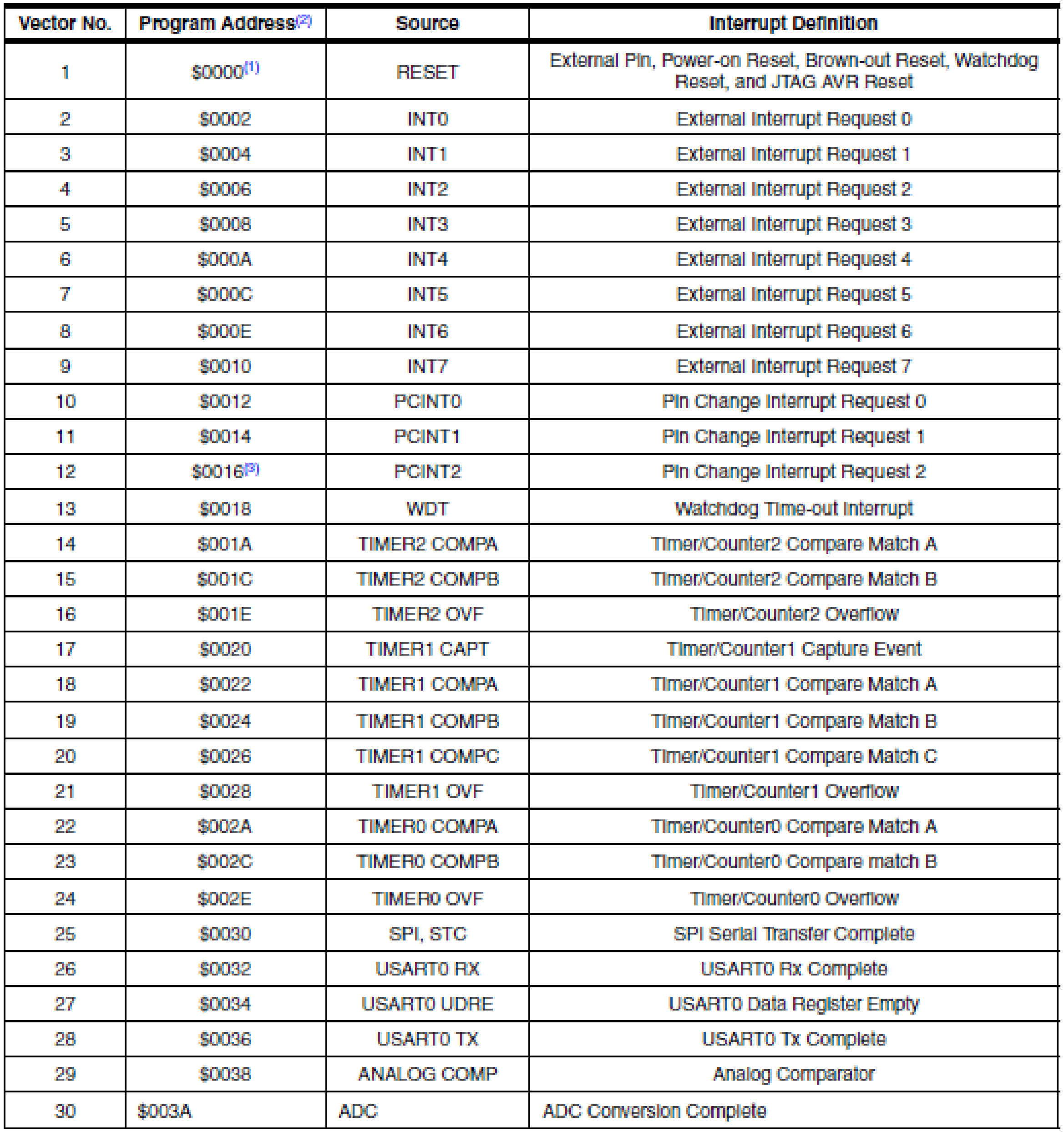


Table 10-1: Interrupt Vector Table for the Mega32

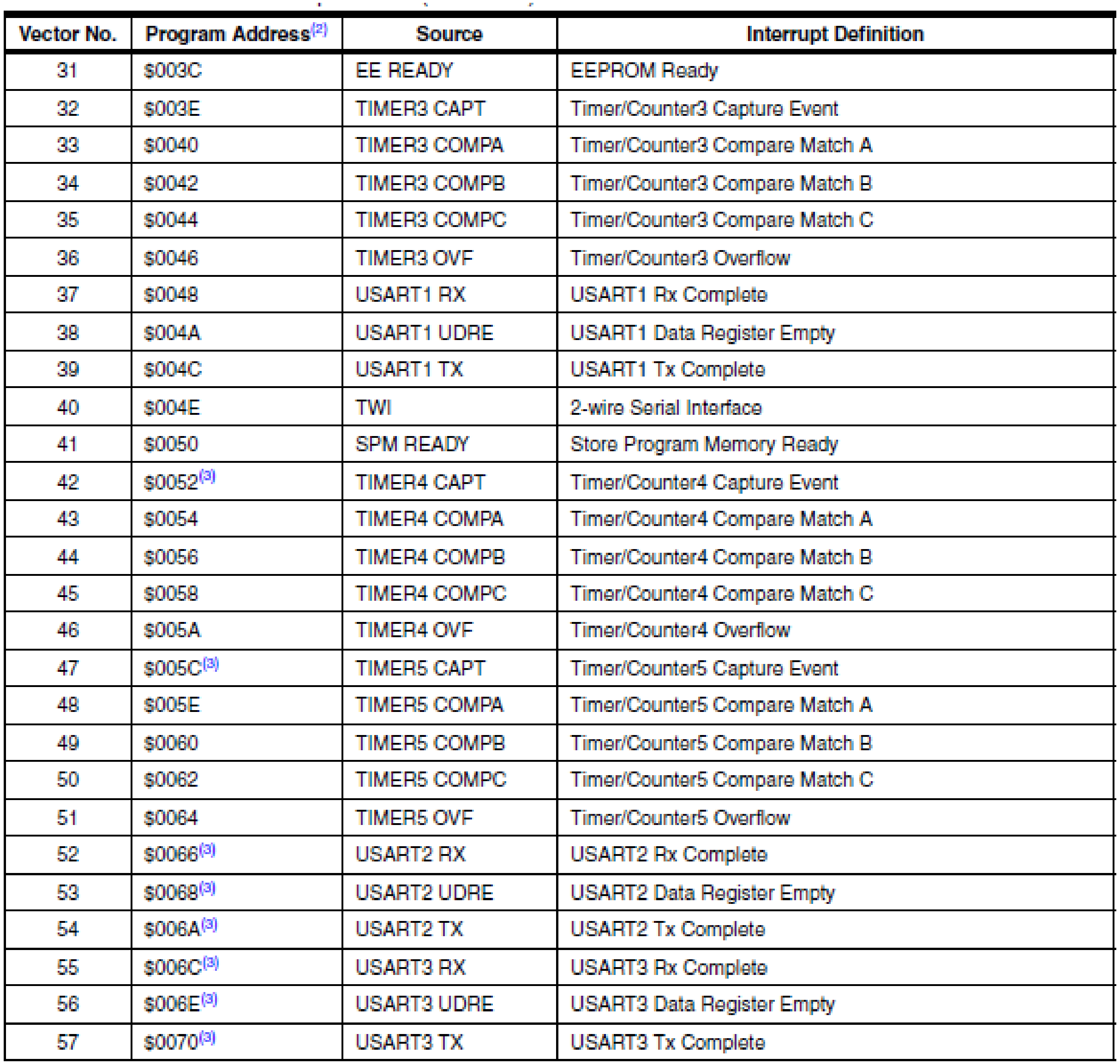
Interrupt	ROM Location (Hex)
Reset	0000
External Interrupt request 0	0002
External Interrupt request 1	0004
External Interrupt request 2	0006
Time/Counter2 Compare Match	0008
Time/Counter2 Overflow	000A
Time/Counter1 Capture Event	000C
Time/Counter1 Compare Match A	000E
Time/Counter1 Compare Match B	0010
Time/Counter1 Overflow	0012
Time/Counter0 Compare Match	0014
Time/Counter0 Overflow	0016
SPI Transfer complete	0018
USART, Receive complete	001A
USART, Data Register Empty	001C
USART, Transmit Complete	001E
ADC Conversion complete	0020
EEPROM ready	0022
Analog Comparator	0024
Two-wire Serial Interface	0026
Store Program Memory Ready	0028

Mega2560: Interrupt vektorer (1 af 2)



Vector No.	Program Address[2]	Source	Interrupt Definition
1	$0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	PCINT0	Pin Change Interrupt Request 0
11	$0014	PCINT1	Pin Change Interrupt Request 1
12	$0016[3]	PCINT2	Pin Change Interrupt Request 2
13	$0018	WDT	Watchdog Time-out Interrupt
14	$001A	TIMER2 COMPA	Timer/Counter2 Compare Match A
15	$001C	TIMER2 COMPB	Timer/Counter2 Compare Match B
16	$001E	TIMER2 OVF	Timer/Counter2 Overflow
17	$0020	TIMER1 CAPT	Timer/Counter1 Capture Event
18	$0022	TIMER1 COMPA	Timer/Counter1 Compare Match A
19	$0024	TIMER1 COMPB	Timer/Counter1 Compare Match B
20	$0026	TIMER1 COMPC	Timer/Counter1 Compare Match C
21	$0028	TIMER1 OVF	Timer/Counter1 Overflow
22	$002A	TIMER0 COMPA	Timer/Counter0 Compare Match A
23	$002C	TIMER0 COMPB	Timer/Counter0 Compare match B
24	$002E	TIMER0 OVF	Timer/Counter0 Overflow
25	$0030	SPI, STC	SPI Serial Transfer Complete
26	$0032	USART0 RX	USART0 Rx Complete
27	$0034	USART0 UDRE	USART0 Data Register Empty
28	$0036	USART0 TX	USART0 Tx Complete
29	$0038	ANALOG COMP	Analog Comparator
30	$003A	ADC	ADC Conversion Complete

Mega2560: Interrupt vektorer (2 af 2)



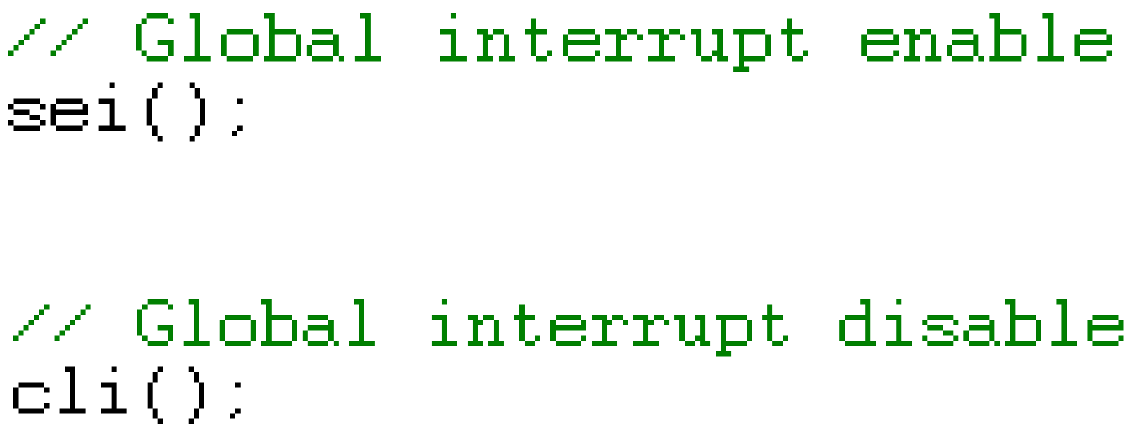
Vector No.	Program Address[2]	Source	Interrupt Definition
31	$003C	EE READY	EEPROM Ready
32	$003E	TIMER3 CAPT	Timer/Counter3 Capture Event
33	$0040	TIMER3 COMPA	Timer/Counter3 Compare Match A
34	$0042	TIMER3 COMPB	Timer/Counter3 Compare Match B
35	$0044	TIMER3 COMPC	Timer/Counter3 Compare Match C
36	$0046	TIMER3 OVF	Timer/Counter3 Overflow
37	$0048	USART1 RX	USART1 Rx Complete
38	$004A	USART1 UDRE	USART1 Data Register Empty
39	$004C	USART1 TX	USART1 Tx Complete
40	$004E	TWI	2-wire Serial Interface
41	$0050	SPM READY	Store Program Memory Ready
42	$0052[3]	TIMER4 CAPT	Timer/Counter4 Capture Event
43	$0054	TIMER4 COMPA	Timer/Counter4 Compare Match A
44	$0056	TIMER4 COMPB	Timer/Counter4 Compare Match B
45	$0058	TIMER4 COMPC	Timer/Counter4 Compare Match C
46	$005A	TIMER4 OVF	Timer/Counter4 Overflow
47	$005C[3]	TIMER5 CAPT	Timer/Counter5 Capture Event
48	$005E	TIMER5 COMPA	Timer/Counter5 Compare Match A
49	$0060	TIMER5 COMPB	Timer/Counter5 Compare Match B
50	$0062	TIMER5 COMPC	Timer/Counter5 Compare Match C
51	$0064	TIMER5 OVF	Timer/Counter5 Overflow
52	$0066[3]	USART2 RX	USART2 Rx Complete
53	$0068[3]	USART2 UDRE	USART2 Data Register Empty
54	$006A[3]	USART2 TX	USART2 Tx Complete
55	$006C[3]	USART3 RX	USART3 Rx Complete
56	$006E[3]	USART3 UDRE	USART3 Data Register Empty
57	$0070[3]	USART3 TX	USART3 Tx Complete

Global interrupt enable

- Via et enkelt bit (Global Interrupt Enable Flag) tillades/forbydes **alle interrupts**.
  Kan betragtes som interruptsystemets hovedafbryder.

- Assembly-instruktionen ”sei” tillader (enabler) interrupts.
- Assembly-instruktionen ”cli” slukker (disabler) interrupts.

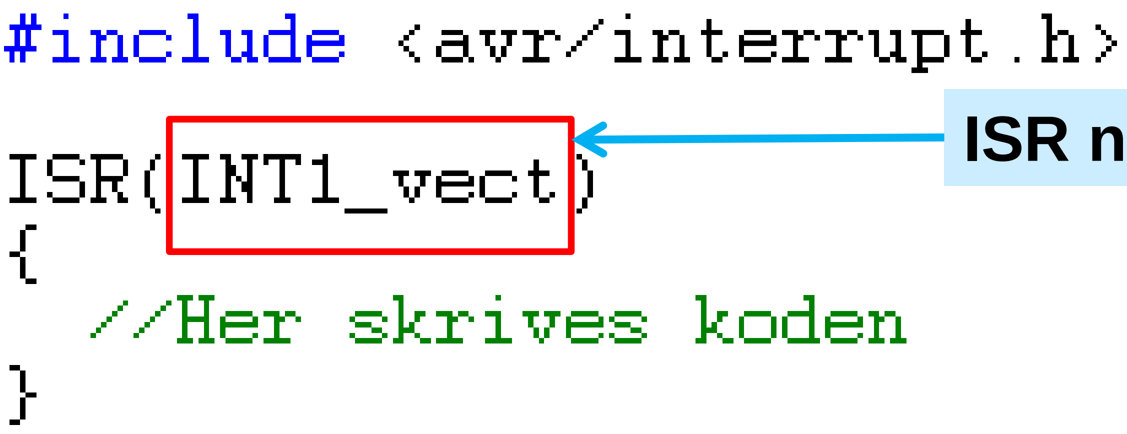
Kode – eksempler (AVR GCC)



```
// Global interrupt enable
sei();


// Global interrupt disable
cli();
```

Kode-eksempel (ISR)



```
#include <avr/interrupt.h>

ISR(INT1_vect)
{
  //Her skrives koden
}
```

ISR navn (= ”type”)

AVR GCC interrupt navne (Mega32)

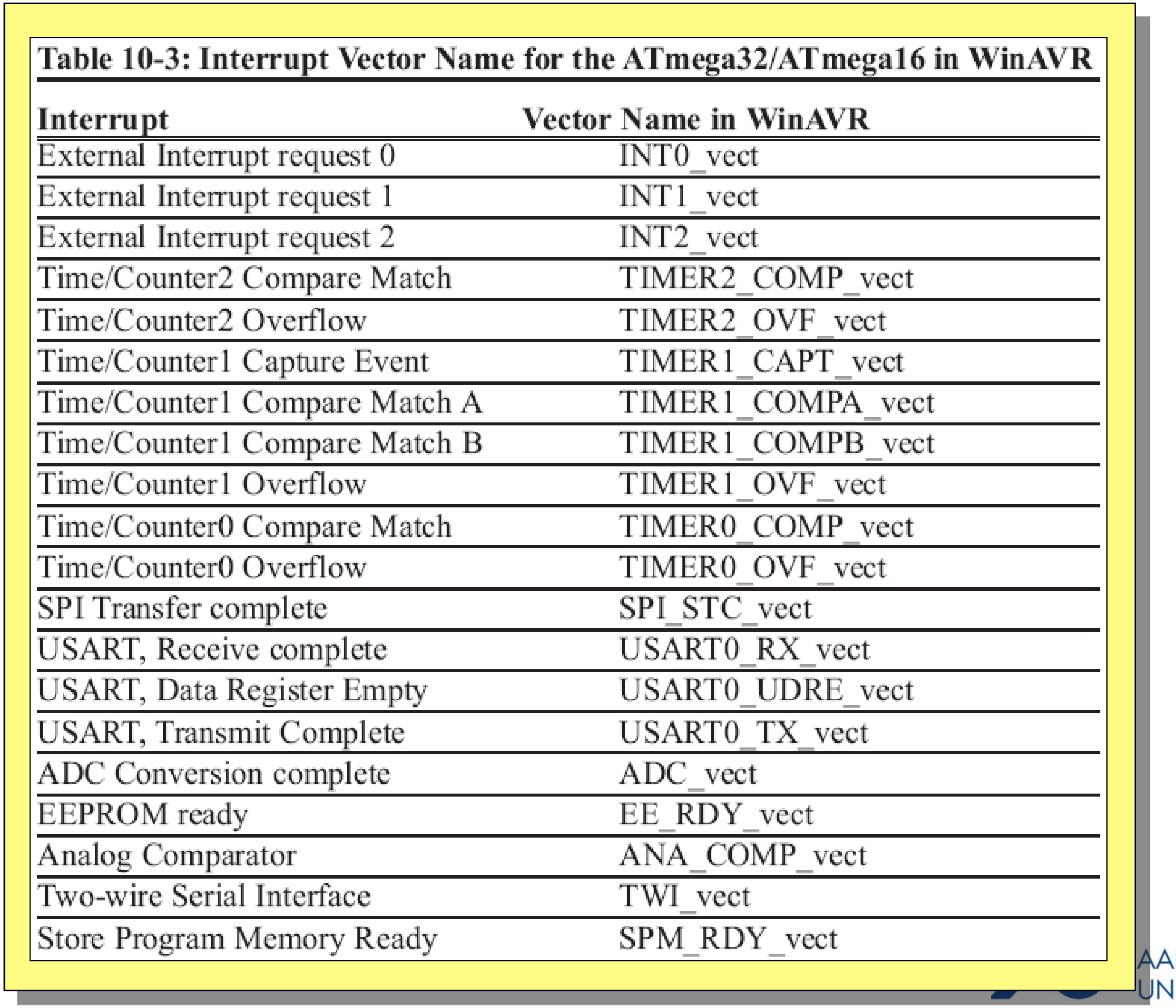


Table 10-3: Interrupt Vector Name for the ATmega32/ATmega16 in WinAVR

Interrupt	Vector Name in WinAVR
External Interrupt request 0	INT0_vect
External Interrupt request 1	INT1_vect
External Interrupt request 2	INT2_vect
Time/Counter2 Compare Match	TIMER2_COMP_vect
Time/Counter2 Overflow	TIMER2_OVF_vect
Time/Counter1 Capture Event	TIMER1_CAPT_vect
Time/Counter1 Compare Match A	TIMER1_COMPA_vect
Time/Counter1 Compare Match B	TIMER1_COMPB_vect
Time/Counter1 Overflow	TIMER1_OVF_vect
Time/Counter0 Compare Match	TIMER0_COMP_vect
Time/Counter0 Overflow	TIMER0_OVF_vect
SPI Transfer complete	SPI_STC_vect
USART, Receive complete	USART0_RX_vect
USART, Data Register Empty	USART0_UDRE_vect
USART, Transmit Complete	USART0_TX_vect
ADC Conversion complete	ADC_vect
EEPROM ready	EE_RDY_vect
Analog Comparator	ANA_COMP_vect
Two-wire Serial Interface	TWI_vect
Store Program Memory Ready	SPM_RDY_vect

Mega2560 interrupt navne (Pins)

External interrupts	ISR name
INT0	INT0_vect
INT1	INT1_vect
INT2	INT2_vect
INT3	INT3_vect
INT4	INT4_vect
INT5	INT5_vect
INT6	INT6_vect
INT7	INT7_vect

PIN change interrupts	ISR name
PCINT0	PCINT0_vect
PCINT1	PCINT1_vect
PCINT2	PCINT2_vect

Mega2560 interrupt navne (Timers)

Timer 0 interrupts	ISR name
COMP A	TIMER0_COMPA_vect
COMP B	TIMER0_COMPB_vect
Overflow	TIMER0_OVF_vect

Timer 1 interrupts	ISR name
Capture	TIMER1_CAPT_vect
COMP A	TIMER1_COMPA_vect
COMP B	TIMER1_COMPB_vect
COMP C	TIMER1_COMPC_vect
Overflow	TIMER1_OVF_vect

Timer 2 interrupts	ISR name
COMP A	TIMER2_COMPA_vect
COMP B	TIMER2_COMPB_vect
Overflow	TIMER2_OVF_vect

Timer 3 interrupts	ISR name
Capture	TIMER3_CAPT_vect
COMP A	TIMER3_COMPA_vect
COMP B	TIMER3_COMPB_vect
COMP C	TIMER3_COMPC_vect
Overflow	TIMER3_OVF_vect

Timer 4 interrupts	ISR name
Capture	TIMER4_CA
COMP A	TIMER4_CO
COMP B	TIMER4_CO
COMP C	TIMER4_CO
Overflow	TIMER4_OV

Timer 5 interrupts	ISR name
Capture	TIMER5_CAPT_vect
COMP A	TIMER5_COMPA_vect
COMP B	TIMER5_COMPB_vect
COMP C	TIMER5_COMPC_vect
Overflow	TIMER5_OVF_vect

Mega2560 interrupt navne (USART)

USART 0 interrupts	ISR name
RX	USART0_RX_vect
TX	USART0_TX_vect
UDR Empty	USART0_UDRE_vect

USART 2 interrupts	ISR name
RX	USART2_RX_vect
TX	USART2_TX_vect
UDR Empty	USART2_UDRE_vect

USART 1 interrupts	ISR name
RX	USART1_RX_vect
TX	USART1_TX_vect
UDR Empty	USART1_UDRE_vect

USART 3 interrupts	ISR name
RX	USART3_RX_vect
TX	USART3_TX_vect
UDR Empty	USART3_UDRE_vect

Mega2560 interrupt navne (diverse)

Other interrupts	ISRname
Watchdog timeout	WDT_vect
SPI	SPI_STC_vect
Analog comparator	ANALOG_COMP_vect
ADC	ADC_vect
EEPROM	EE_READY_vect
TWI	TWI_vect
SPM ready	SPM_READY_vect

Lokale interrupt enables

- **De enkelte interrupttyper** har hver især et **lokalt** interrupt enable flag.

- **Kun** hvis dette lokale interrupt enable flag er sat, **OG** det globale interrupt flag er sat, vil enheden kunne starte et interrupt !

(Mega32: Eksterne interrupts)

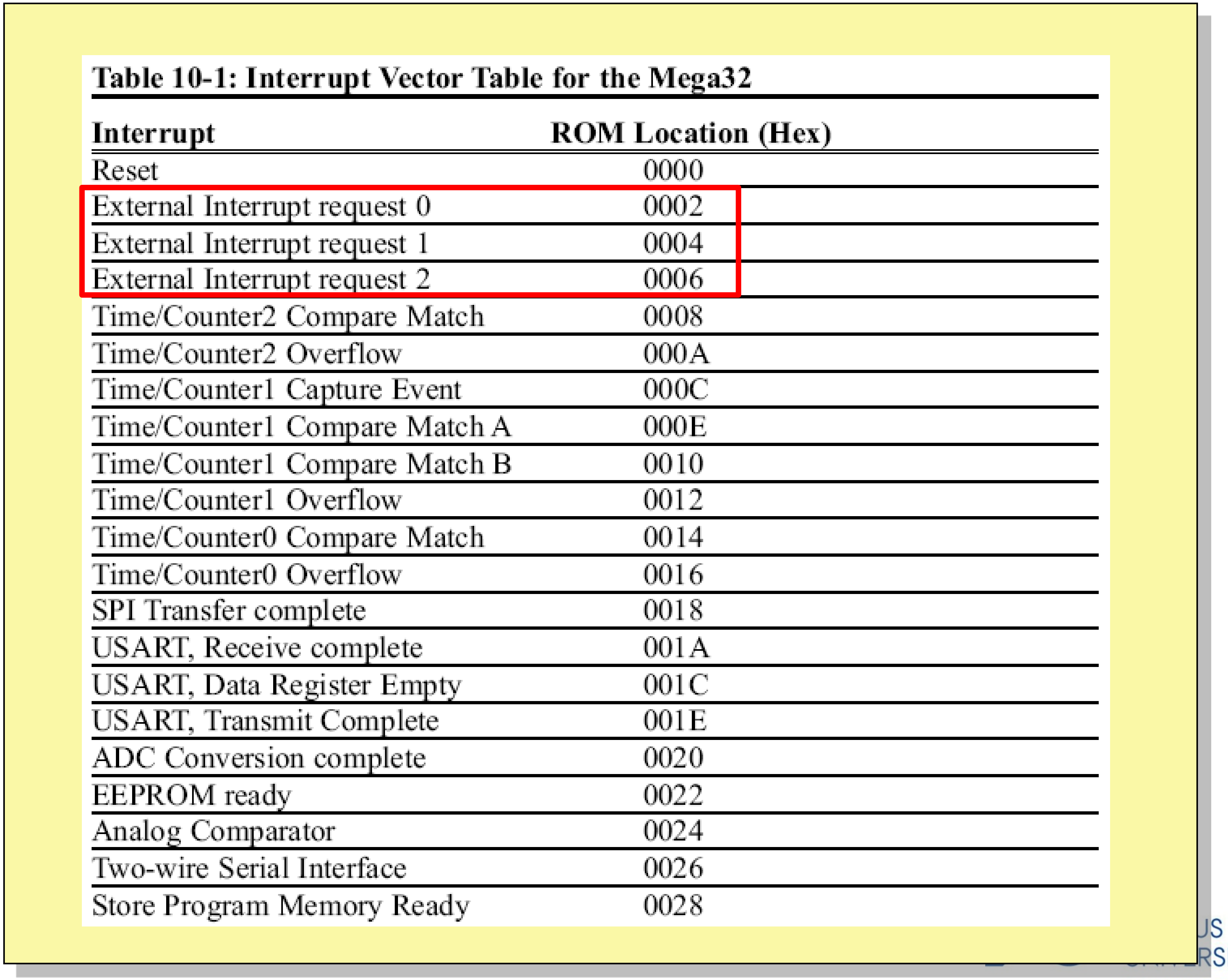
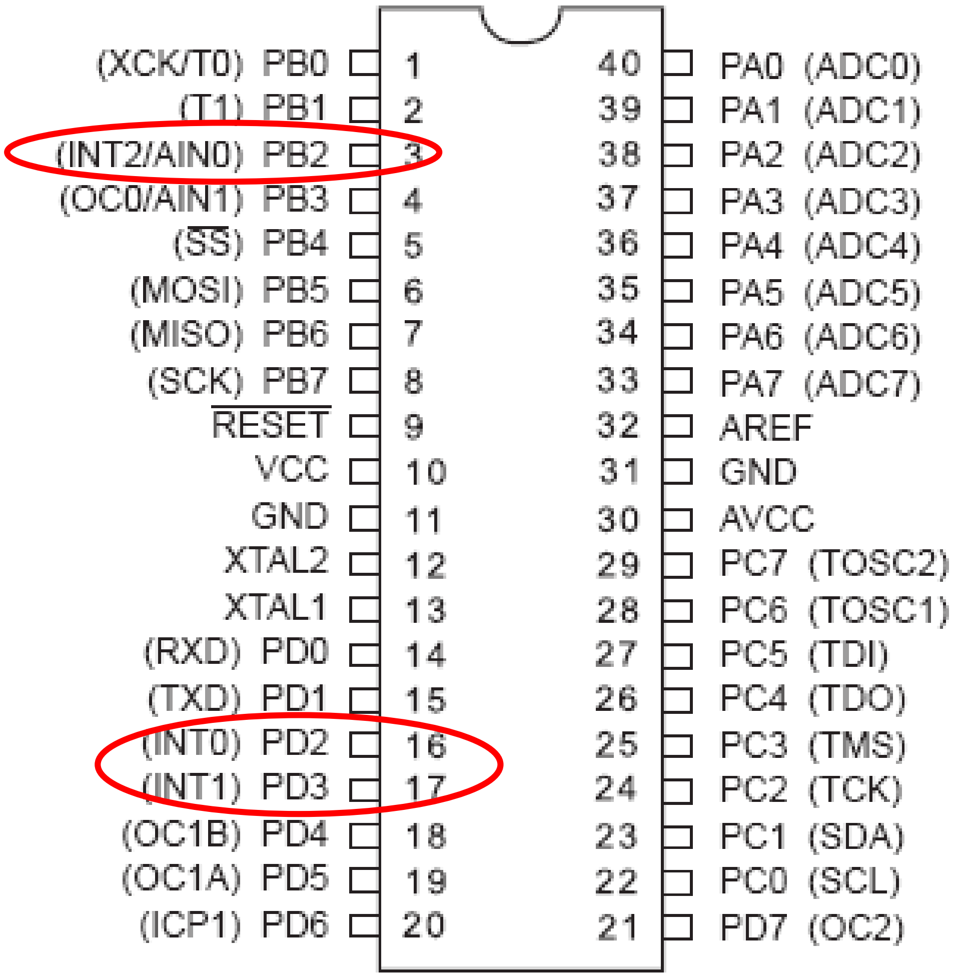


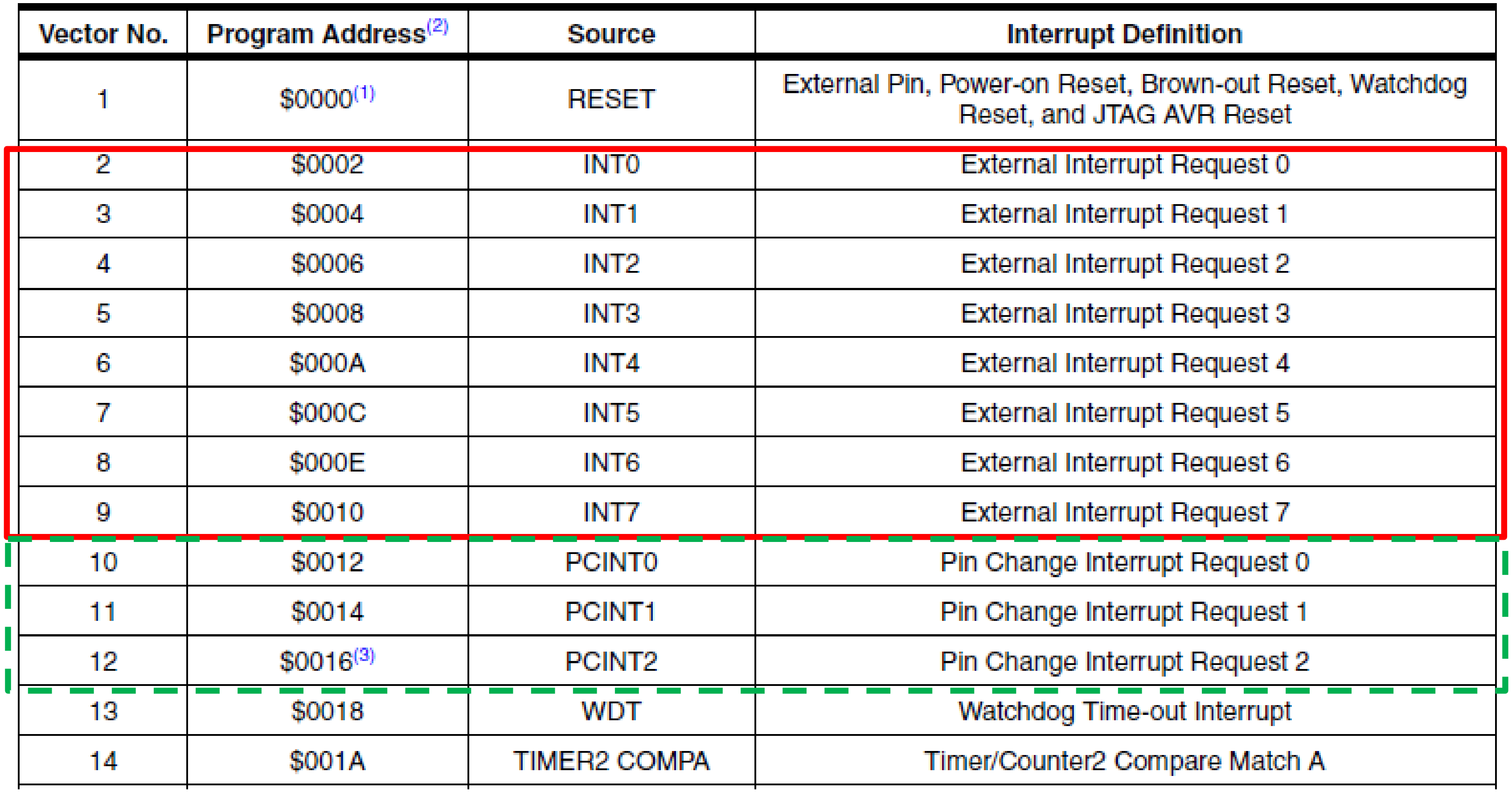
Table 10-1: Interrupt Vector Table for the Mega32

Interrupt	ROM Location (Hex)
Reset	0000
External Interrupt request 0	0002
External Interrupt request 1	0004
External Interrupt request 2	0006
Time/Counter2 Compare Match	0008
Time/Counter2 Overflow	000A
Time/Counter1 Capture Event	000C
Time/Counter1 Compare Match A	000E
Time/Counter1 Compare Match B	0010
Time/Counter1 Overflow	0012
Time/Counter0 Compare Match	0014
Time/Counter0 Overflow	0016
SPI Transfer complete	0018
USART, Receive complete	001A
USART, Data Register Empty	001C
USART, Transmit Complete	001E
ADC Conversion complete	0020
EEPROM ready	0022
Analog Comparator	0024
Two-wire Serial Interface	0026
Store Program Memory Ready	0028

(Mega32: Eksterne interrupt ben)



Mega2560: Eksterne interrupts



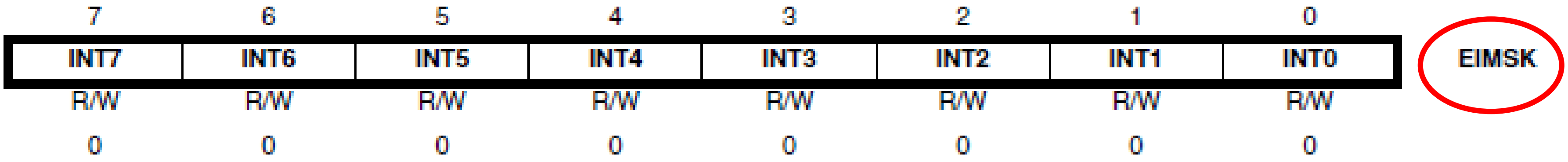
Vector No.	Program Address[2]	Source	Interrupt Definition
1	$0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	PCINT0	Pin Change Interrupt Request 0
11	$0014	PCINT1	Pin Change Interrupt Request 1
12	$0016[3]	PCINT2	Pin Change Interrupt Request 2
13	$0018	WDT	Watchdog Time-out Interrupt
14	$001A	TIMER2 COMPA	Timer/Counter2 Compare Match A

Mega2560: Eksterne interrupt ben

INT0 = PD, ben 0
INT1 = PD, ben 1
INT2 = PD, ben 2 = SW2 på Mega2560 I/O shield
INT3 = PD, ben 3 = SW3 på Mega2560 I/O shield

INT4 = PE, ben 4
INT5 = PE, ben 5
INT6 = PE, ben 6
INT7 = PE, ben 7

Mega2560: Ekstern interrupt enable bits

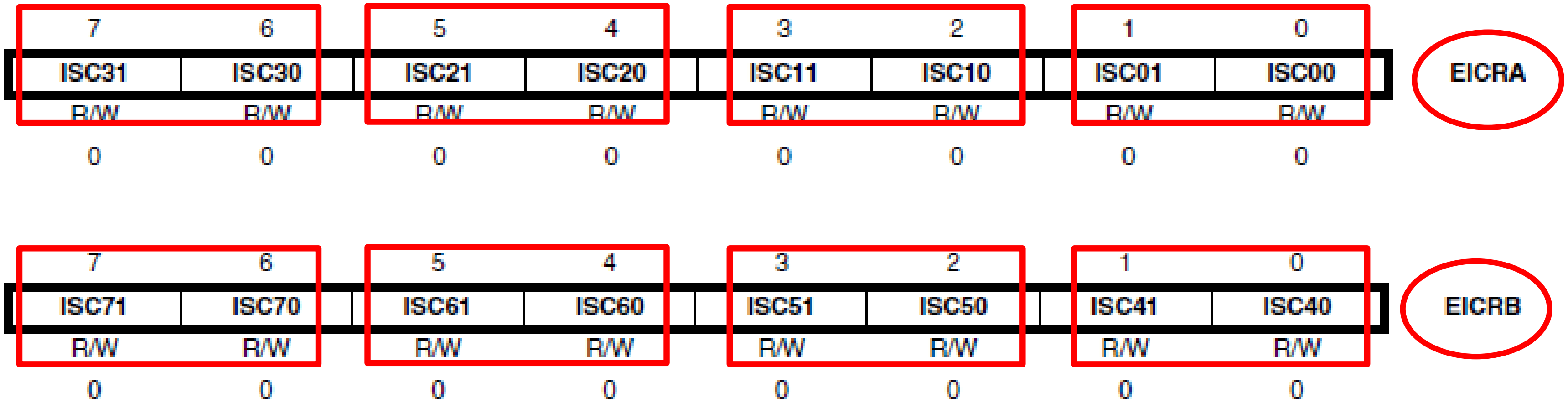


7	6	5	4	3	2	1	0	
INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

- 0 : Disabled.
- 1 : Enabled.

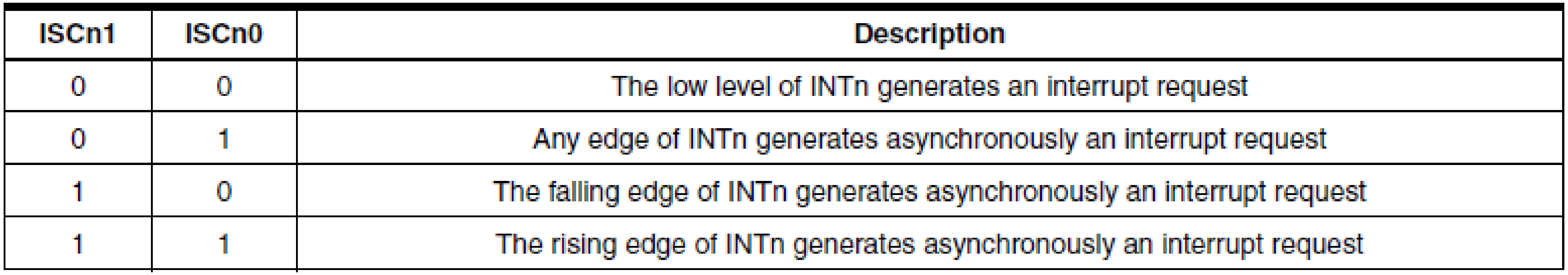
- Kodeeksempel :
  EIMSK = EIMSK | 0b00001100;
  -- eller --
  EIMSK |= 0b00001100;

Mega2560: Trignings-muligheder, INT0-INT7



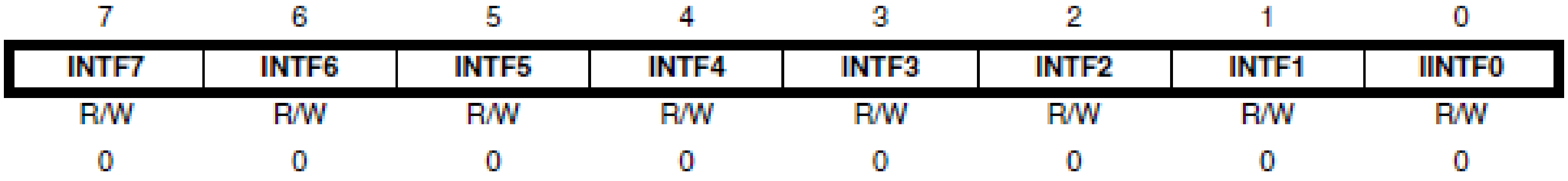
7	6	5	4	3	2	1	0	
ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	
ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICRB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	



ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request
0	1	Any edge of INTn generates asynchronously an interrupt request
1	0	The falling edge of INTn generates asynchronously an interrupt request
1	1	The rising edge of INTn generates asynchronously an interrupt request

(Mega2560: Eksterne interrupts: Flags)



7	6	5	4	3	2	1	0
INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	IINTF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

EIFR

- Sættes, når interrupt trigges (FLIP-FLOP).
- Hvis globalt interrupt enabled og lokalt enabled: Interrupt starter herefter.
- Flag nulstilles automatisk i interrupt rutinen (eller af SW ved at skrive 1 til det aktuelle bit).
- Det er sjældent, at vi har behov for at bruge dette register (når interrupt er enabled).

Mega2560: Pin change interupts (PCI)

- Mega2560 har desuden mulighed for noget, der kaldes ”Pin Change Interrupts” (der har Mega32 ikke).
- Princippet er, at vi vil kunne få interrupt, hvis et (ud af i alt 24) ben skifter tilstand (low->high eller high->low).
- PCI behandles ikke i dette kursus, men er naturligvis beskrevet i Mega2560 databogen (findes på MSYS Blackboard).

Test (”socrative.com”: Room = MSYS)

Vi ønsker at enable Mega2560's eksterne interrupts INT0 og INT4 (de øvrige interrupts skal være disabled). INT0 skal trigge på "falling edge" og INT4 skal trigge på "rising edge". Hvilken kode er korrekt initiering ?

- A: EICRA = 0b00000010;
  EICRB = 0b00000011;
  EIMSK = 0b00010001;
- B: EICRA = 0b00000011;
  EICRB = 0b00000010;
  EIMSK = 0b00010001;
- C: EICRA = 0b00000010;
  EICRB = 0b00000000;
  EIMSK = 0b10000001;

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Test (”socrative.com”: Room = MSYS)

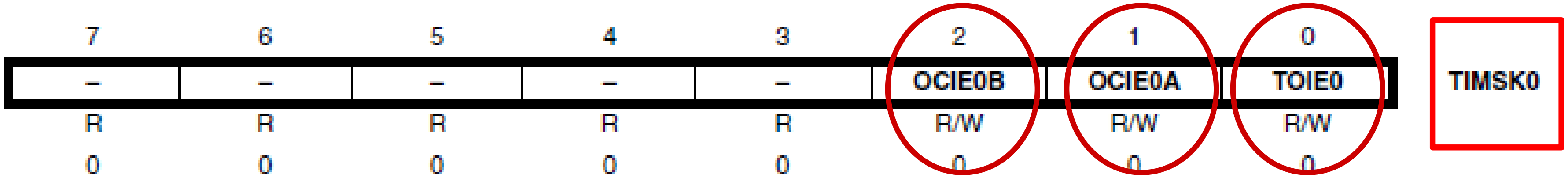
For Mega2560 er det eksterne interrupt INT0 er initieret på følgende måde:

```
EICRA = 0b00000001;
EIMSK |= 0b00000001;
sei( );
```

Hvilken hændelse på INT0 benet vil starte et interrupt ?

- A: Når signalet et lavt (0 volt).
- B: Når signalet ændrer tilstand ("any logical change").
- C: Når signalet går fra høj til lav ("falling edge").
- D: Når signalet går fra lav til høj ("rising edge");

Mega2560: Timer 0 interrupt enables



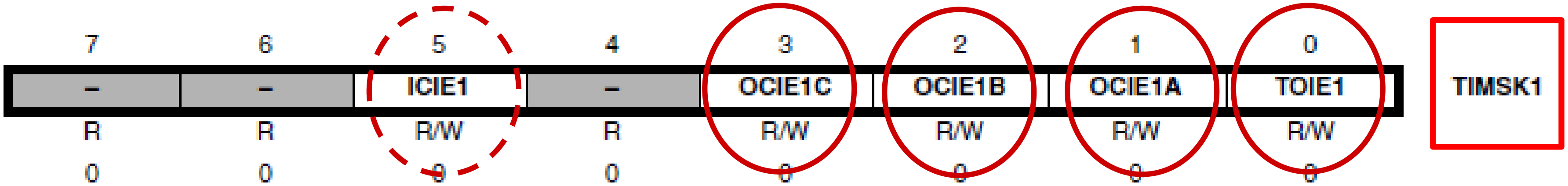
7	6	5	4	3	2	1	0	
–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
R	R	R	R	R	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

- Bit TOIE0 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.

- Bit OCIE0A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.

- Bit OCIE0B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

Mega2560: Timer 1 interrupt enables



7	6	5	4	3	2	1	0	
–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
R	R	R/W	R	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

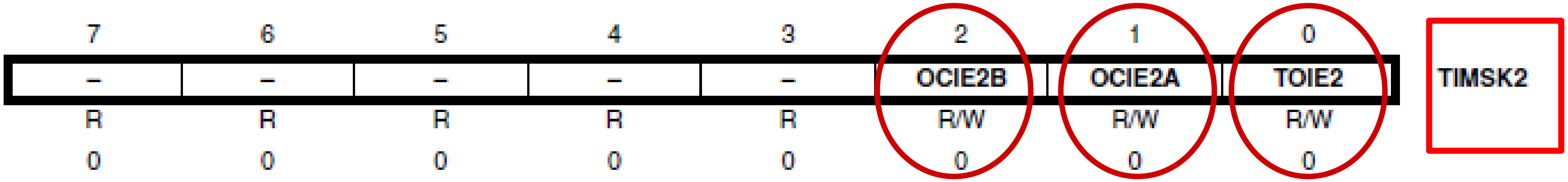
- Bit TOIE1 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.

- Bit OCIE1A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.

- Bit OCIE1B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

- Bit OCIE1C (bit 3):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

Mega2560: Timer 2 interrupt enables



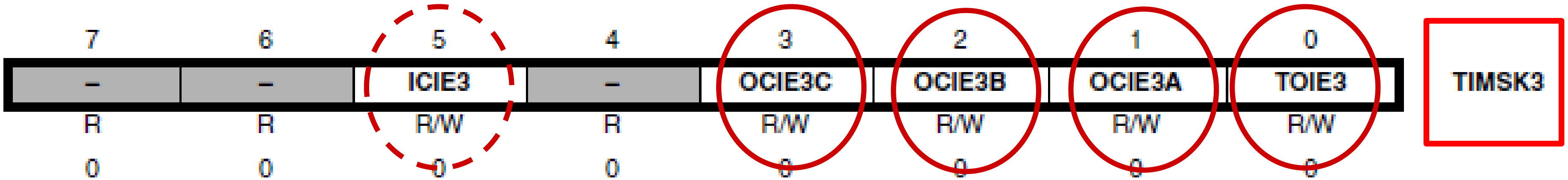
7	6	5	4	3	2	1	0	
–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	TIMSK2
R	R	R	R	R	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

- Bit TOIE2 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.

- Bit OCIE2A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.

- Bit OCIE2B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

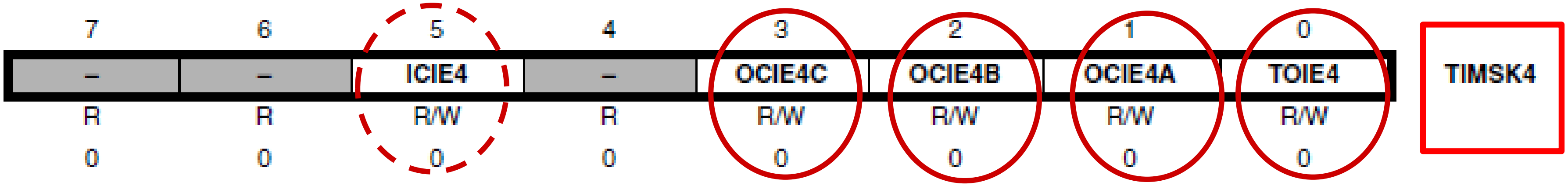
Mega2560: Timer 3 interrupt enables



Bit	7	6	5	4	3	2	1	0	
	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
	R	R	R/W	R	R/W	R/W	R/W	R/W	
	0	0	0	0	0	0	0	0	

- Bit TOIE3 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.
- Bit OCIE3A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.
- Bit OCIE3B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.
- Bit OCIE3C (bit 3):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

Mega2560: Timer 4 interrupt enables



7	6	5	4	3	2	1	0	
–	–	ICIE4	–	OCIE4C	OCIE4B	OCIE4A	TOIE4	TIMSK4
R	R	R/W	R	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

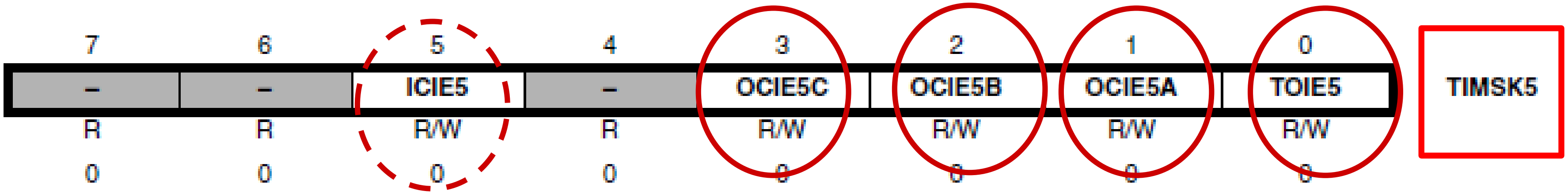
- Bit TOIE4 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.

- Bit OCIE4A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.

- Bit OCIE4B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

- Bit OCIE4C (bit 3):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

Mega2560: Timer 5 interrupt enables



7	6	5	4	3	2	1	0	
–	–	ICIE5	–	OCIE5C	OCIE5B	OCIE5A	TOIE5	TIMSK5
R	R	R/W	R	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

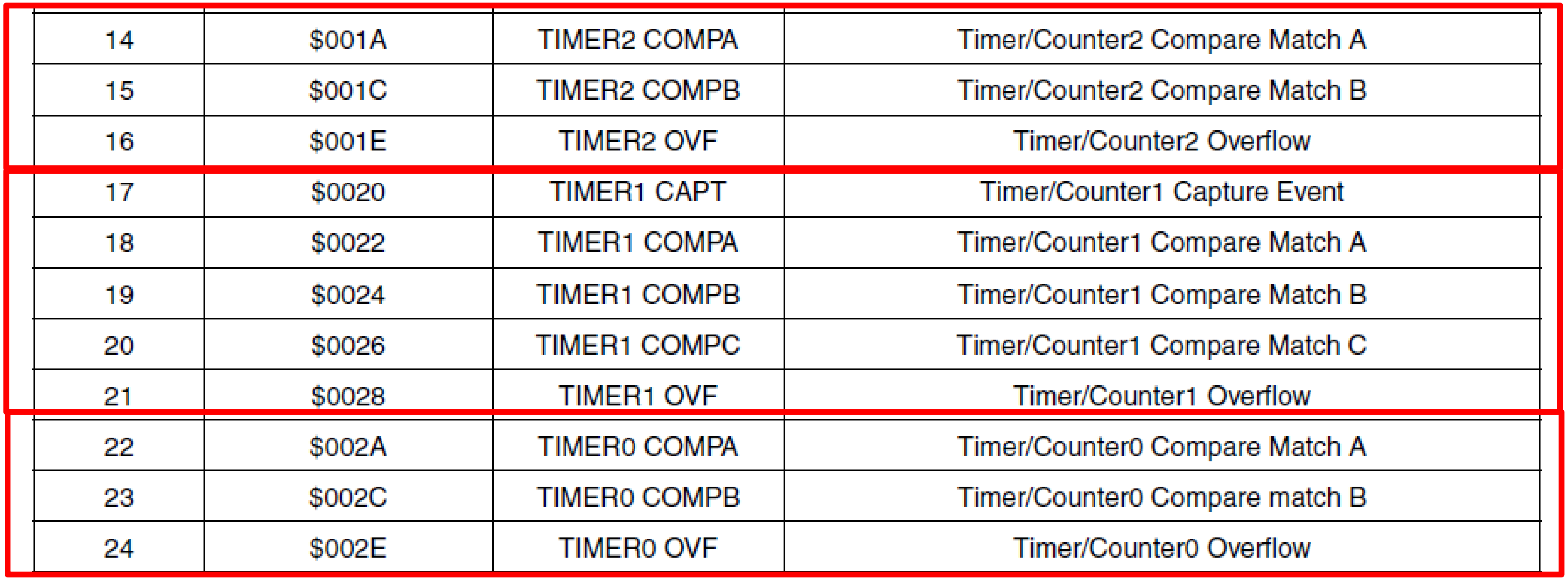
- Bit TOIE5 (bit 0):
  0 => Disable overflow interrupt.
  1 => Enable overflow interrupt.

- Bit OCIE5A (bit 1):
  0 => Disable output compare interrupt, A-systemet.
  1 => Enable output compare interrupt, A-systemet.

- Bit OCIE5B (bit 2):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

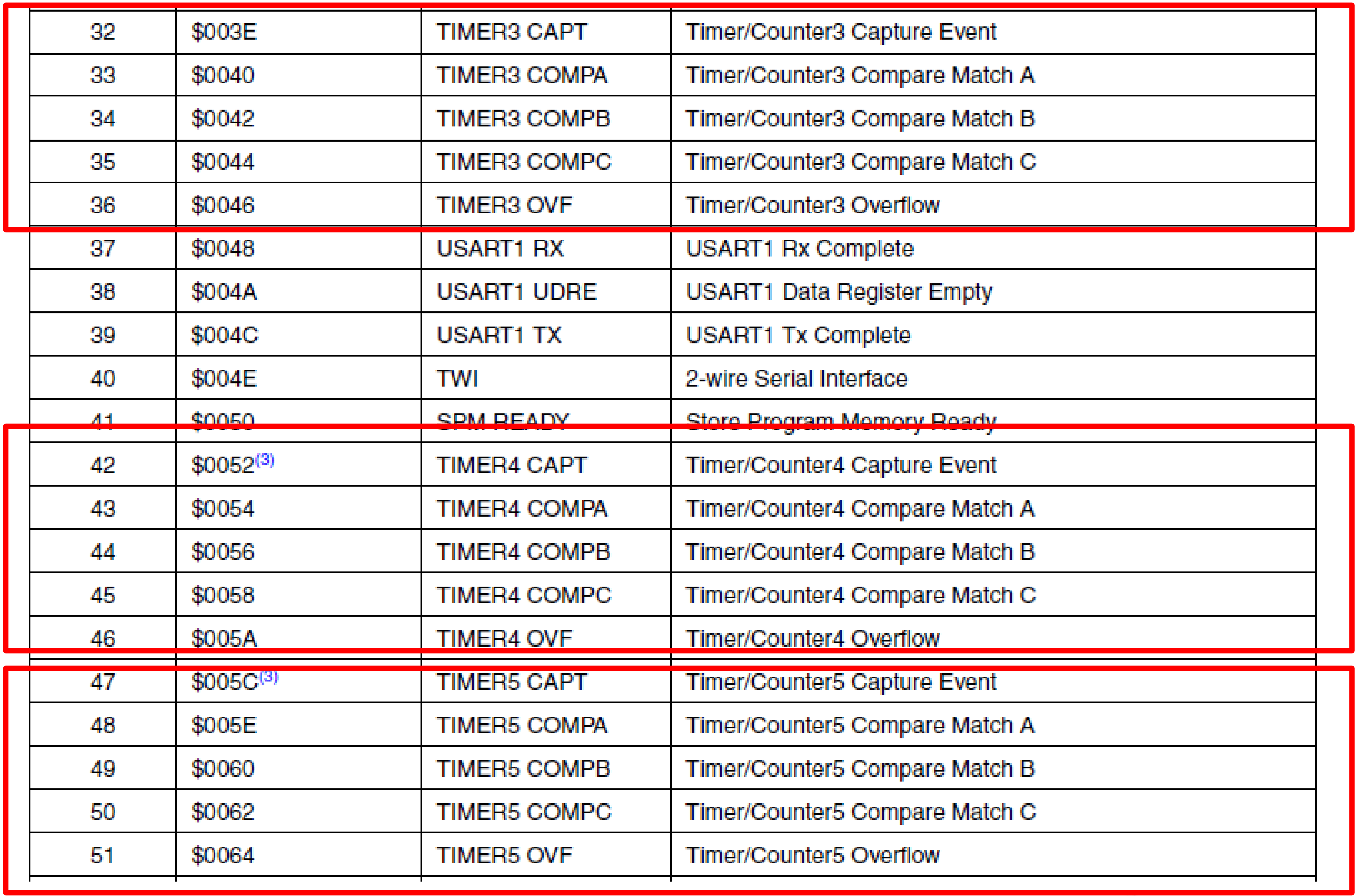
- Bit OCIE5C (bit 3):
  0 => Disable output compare interrupt, B-systemet.
  1 => Enable output compare interrupt, B-systemet.

Mega2560: Timer 0, 1 og 2 interrupts



14	$001A	TIMER2 COMPA	Timer/Counter2 Compare Match A
15	$001C	TIMER2 COMPB	Timer/Counter2 Compare Match B
16	$001E	TIMER2 OVF	Timer/Counter2 Overflow
17	$0020	TIMER1 CAPT	Timer/Counter1 Capture Event
18	$0022	TIMER1 COMPA	Timer/Counter1 Compare Match A
19	$0024	TIMER1 COMPB	Timer/Counter1 Compare Match B
20	$0026	TIMER1 COMPC	Timer/Counter1 Compare Match C
21	$0028	TIMER1 OVF	Timer/Counter1 Overflow
22	$002A	TIMER0 COMPA	Timer/Counter0 Compare Match A
23	$002C	TIMER0 COMPB	Timer/Counter0 Compare match B
24	$002E	TIMER0 OVF	Timer/Counter0 Overflow

Mega2560: Timer 3, 4 og 5 interrupts



32	$003E	TIMER3 CAPT	Timer/Counter3 Capture Event
33	$0040	TIMER3 COMPA	Timer/Counter3 Compare Match A
34	$0042	TIMER3 COMPB	Timer/Counter3 Compare Match B
35	$0044	TIMER3 COMPC	Timer/Counter3 Compare Match C
36	$0046	TIMER3 OVF	Timer/Counter3 Overflow
37	$0048	USART1 RX	USART1 Rx Complete
38	$004A	USART1 UDRE	USART1 Data Register Empty
39	$004C	USART1 TX	USART1 Tx Complete
40	$004E	TWI	2-wire Serial Interface
41	$0050	SPM READY	Store Program Memory Ready
42	$0052[3]	TIMER4 CAPT	Timer/Counter4 Capture Event
43	$0054	TIMER4 COMPA	Timer/Counter4 Compare Match A
44	$0056	TIMER4 COMPB	Timer/Counter4 Compare Match B
45	$0058	TIMER4 COMPC	Timer/Counter4 Compare Match C
46	$005A	TIMER4 OVF	Timer/Counter4 Overflow
47	$005C[3]	TIMER5 CAPT	Timer/Counter5 Capture Event
48	$005E	TIMER5 COMPA	Timer/Counter5 Compare Match A
49	$0060	TIMER5 COMPB	Timer/Counter5 Compare Match B
50	$0062	TIMER5 COMPC	Timer/Counter5 Compare Match C
51	$0064	TIMER5 OVF	Timer/Counter5 Overflow

Test (”socrative.com”: Room = MSYS)

Hvordan enables Timer 1 overflow interrupts for Mega2560 ?

- A: TIMSK |= 0b00000001;
  sei();
- B: TIMSK1 |= 0b00000001;
  sei();
- C: TIMSK1 |= 0b01000000;
  sei();
- D: TIMSK &= 0b00000001;
  sei();

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

(Mega32: Interrupt prioritering)

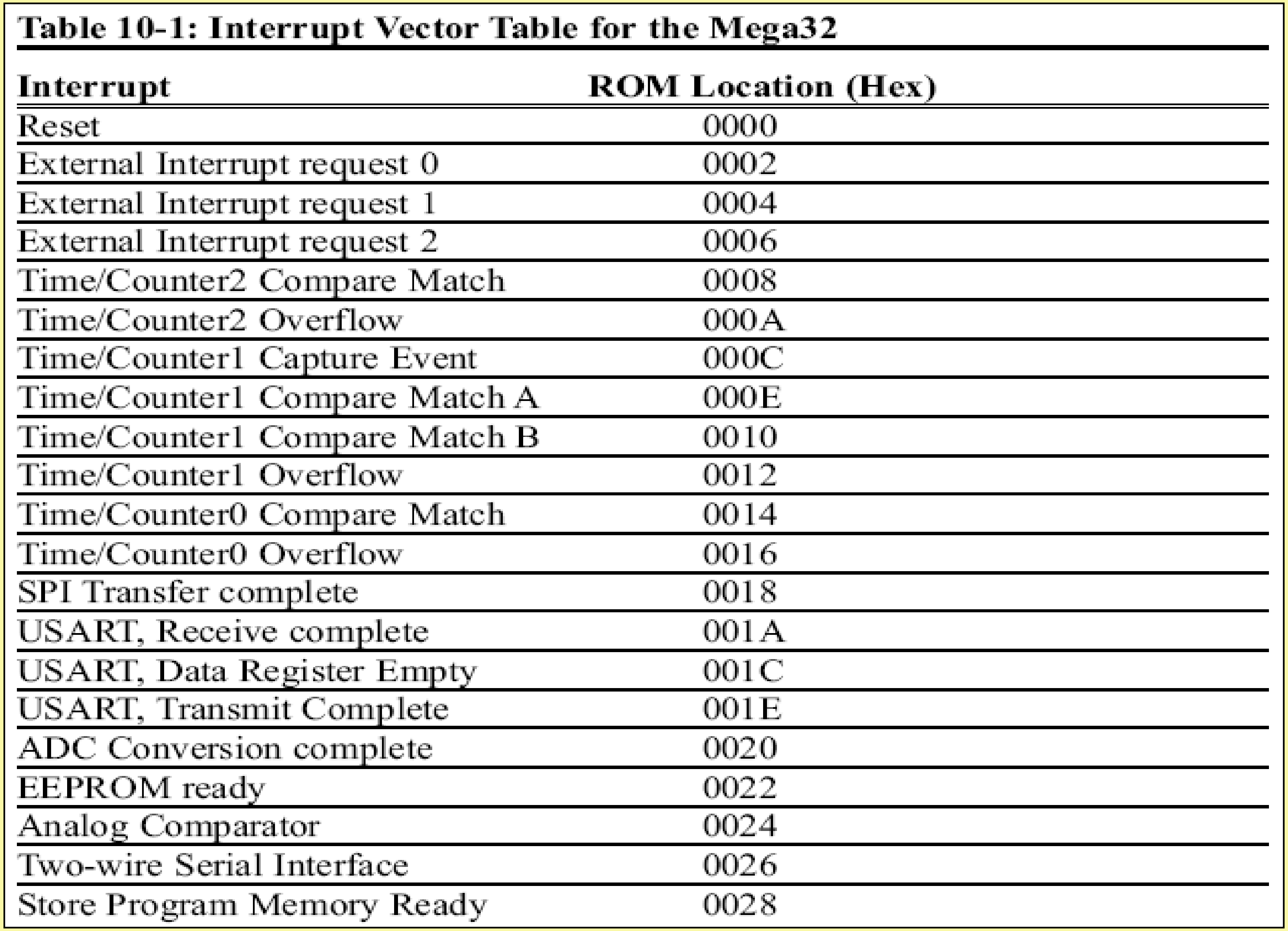
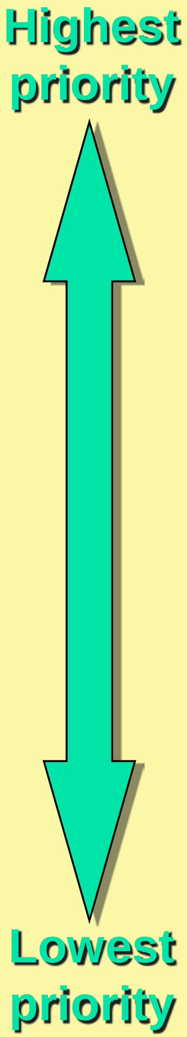


Table 10-1: Interrupt Vector Table for the Mega32

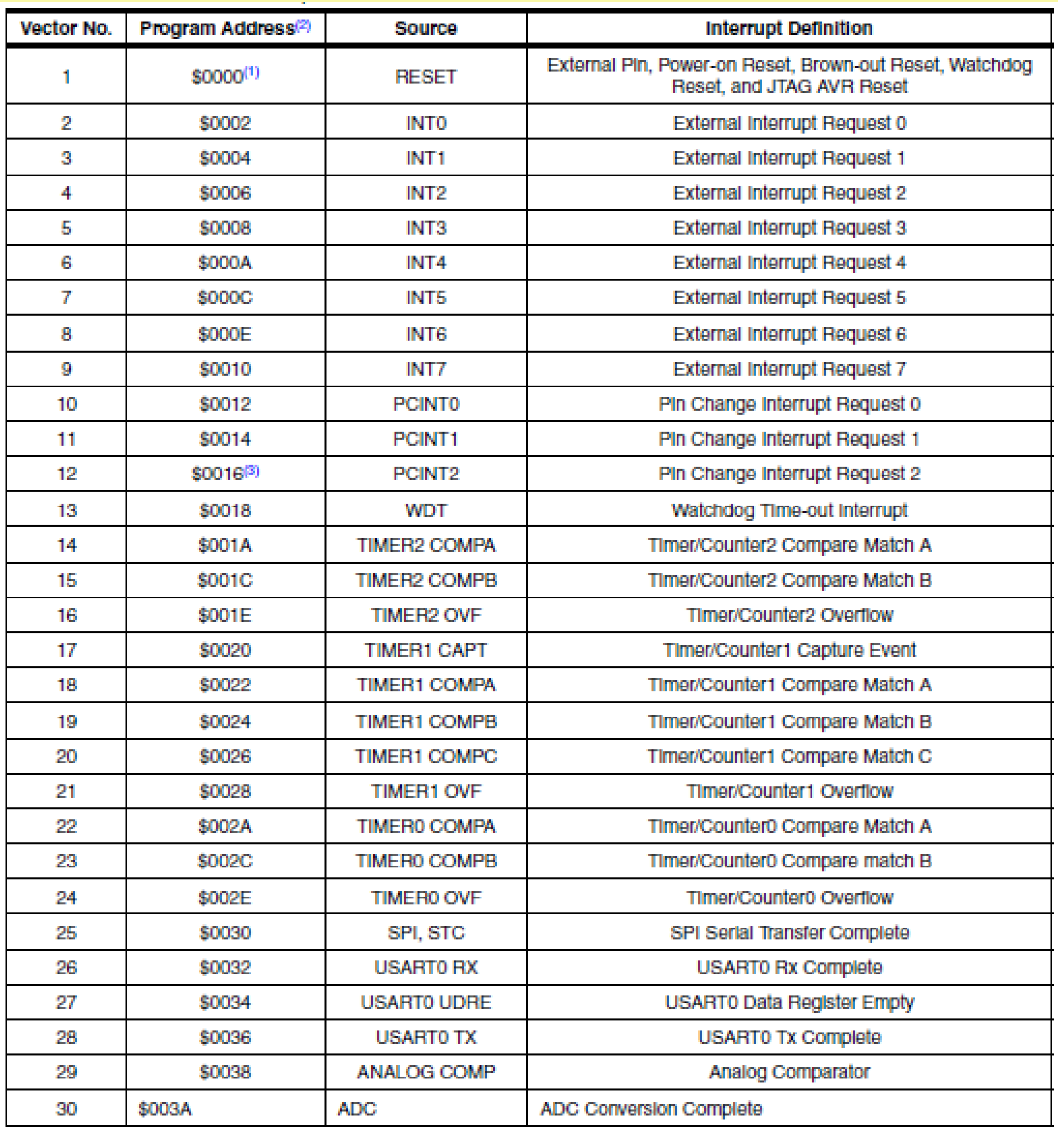
Interrupt	ROM Location (Hex)
Reset	0000
External Interrupt request 0	0002
External Interrupt request 1	0004
External Interrupt request 2	0006
Time/Counter2 Compare Match	0008
Time/Counter2 Overflow	000A
Time/Counter1 Capture Event	000C
Time/Counter1 Compare Match A	000E
Time/Counter1 Compare Match B	0010
Time/Counter1 Overflow	0012
Time/Counter0 Compare Match	0014
Time/Counter0 Overflow	0016
SPI Transfer complete	0018
USART, Receive complete	001A
USART, Data Register Empty	001C
USART, Transmit Complete	001E
ADC Conversion complete	0020
EEPROM ready	0022
Analog Comparator	0024
Two-wire Serial Interface	0026
Store Program Memory Ready	0028



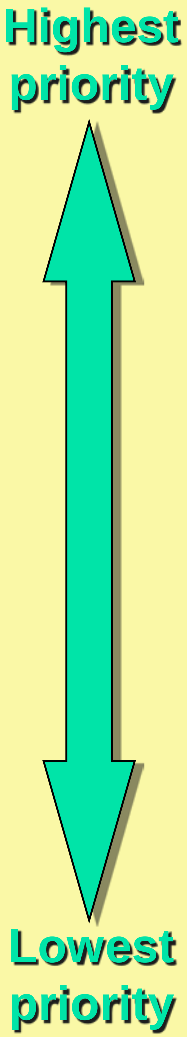
Highest priority

Lowest priority

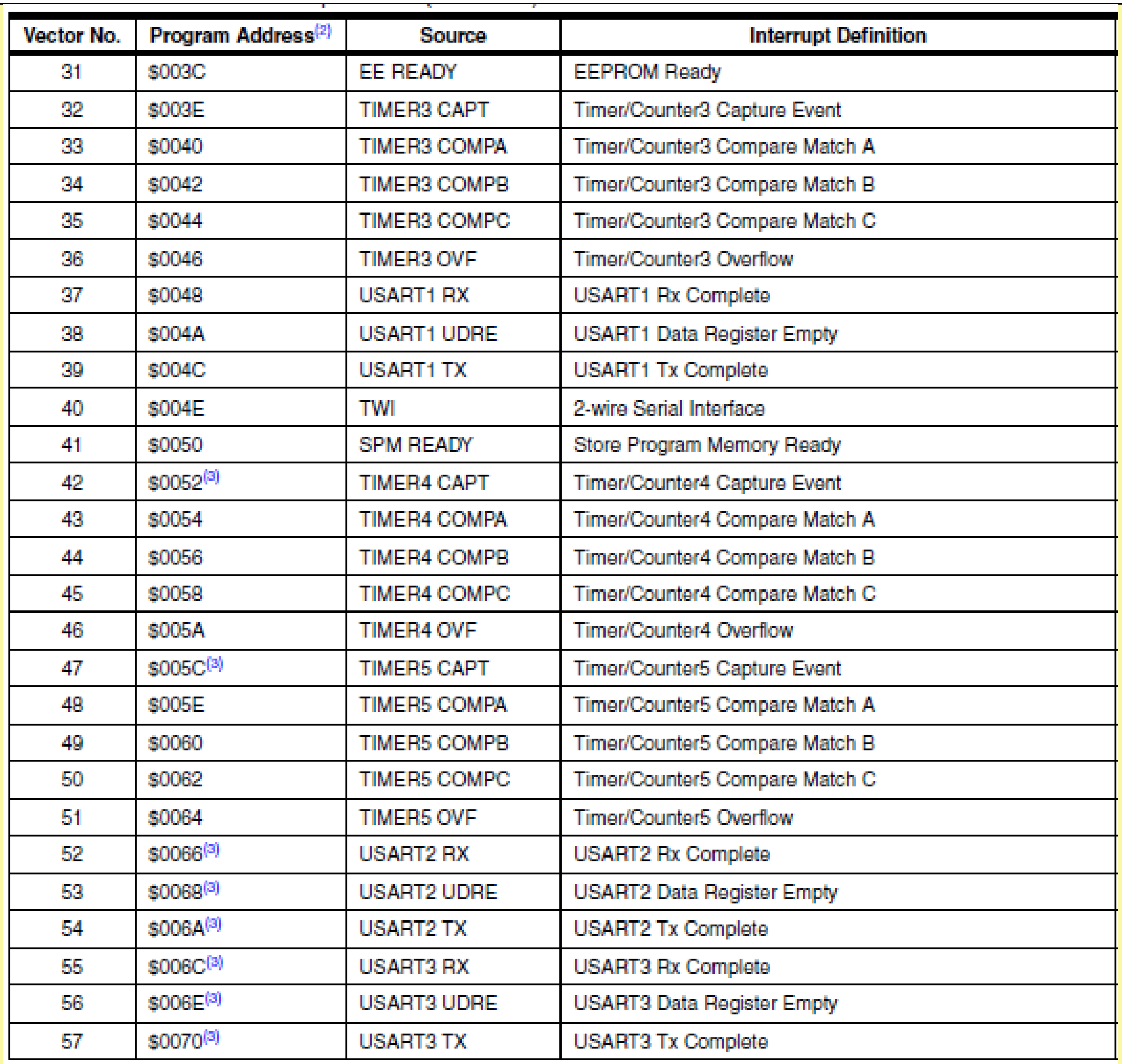
Mega2560: Interrupt prioritering (1 af 2)



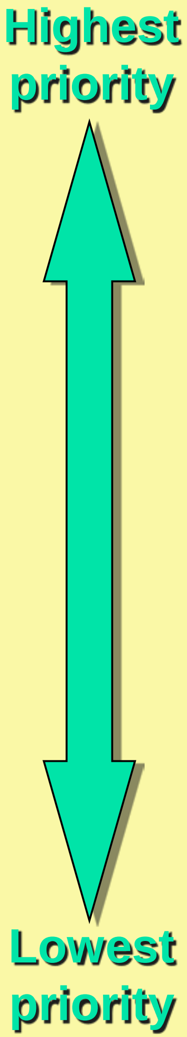
Vector No.	Program Address[2]	Source	Interrupt Definition
1	$0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	PCINT0	Pin Change Interrupt Request 0
11	$0014	PCINT1	Pin Change Interrupt Request 1
12	$0016[3]	PCINT2	Pin Change Interrupt Request 2
13	$0018	WDT	Watchdog Time-out Interrupt
14	$001A	TIMER2 COMPA	Timer/Counter2 Compare Match A
15	$001C	TIMER2 COMPB	Timer/Counter2 Compare Match B
16	$001E	TIMER2 OVF	Timer/Counter2 Overflow
17	$0020	TIMER1 CAPT	Timer/Counter1 Capture Event
18	$0022	TIMER1 COMPA	Timer/Counter1 Compare Match A
19	$0024	TIMER1 COMPB	Timer/Counter1 Compare Match B
20	$0026	TIMER1 COMPC	Timer/Counter1 Compare Match C
21	$0028	TIMER1 OVF	Timer/Counter1 Overflow
22	$002A	TIMER0 COMPA	Timer/Counter0 Compare Match A
23	$002C	TIMER0 COMPB	Timer/Counter0 Compare match B
24	$002E	TIMER0 OVF	Timer/Counter0 Overflow
25	$0030	SPI, STC	SPI Serial Transfer Complete
26	$0032	USART0 RX	USART0 Rx Complete
27	$0034	USART0 UDRE	USART0 Data Register Empty
28	$0036	USART0 TX	USART0 Tx Complete
29	$0038	ANALOG COMP	Analog Comparator
30	$003A	ADC	ADC Conversion Complete



Mega2560: Interrupt prioritering (2 af 2)



Vector No.	Program Address[2]	Source	Interrupt Definition
31	$003C	EE READY	EEPROM Ready
32	$003E	TIMER3 CAPT	Timer/Counter3 Capture Event
33	$0040	TIMER3 COMPA	Timer/Counter3 Compare Match A
34	$0042	TIMER3 COMPB	Timer/Counter3 Compare Match B
35	$0044	TIMER3 COMPC	Timer/Counter3 Compare Match C
36	$0046	TIMER3 OVF	Timer/Counter3 Overflow
37	$0048	USART1 RX	USART1 Rx Complete
38	$004A	USART1 UDRE	USART1 Data Register Empty
39	$004C	USART1 TX	USART1 Tx Complete
40	$004E	TWI	2-wire Serial Interface
41	$0050	SPM READY	Store Program Memory Ready
42	$0052[3]	TIMER4 CAPT	Timer/Counter4 Capture Event
43	$0054	TIMER4 COMPA	Timer/Counter4 Compare Match A
44	$0056	TIMER4 COMPB	Timer/Counter4 Compare Match B
45	$0058	TIMER4 COMPC	Timer/Counter4 Compare Match C
46	$005A	TIMER4 OVF	Timer/Counter4 Overflow
47	$005C[3]	TIMER5 CAPT	Timer/Counter5 Capture Event
48	$005E	TIMER5 COMPA	Timer/Counter5 Compare Match A
49	$0060	TIMER5 COMPB	Timer/Counter5 Compare Match B
50	$0062	TIMER5 COMPC	Timer/Counter5 Compare Match C
51	$0064	TIMER5 OVF	Timer/Counter5 Overflow
52	$0066[3]	USART2 RX	USART2 Rx Complete
53	$0068[3]	USART2 UDRE	USART2 Data Register Empty
54	$006A[3]	USART2 TX	USART2 Tx Complete
55	$006C[3]	USART3 RX	USART3 Rx Complete
56	$006E[3]	USART3 UDRE	USART3 Data Register Empty
57	$0070[3]	USART3 TX	USART3 Tx Complete



Highest priority

Lowest priority

Interrupt inden i et interrupt (“nesting”)

- Det globale interrupt-flag nulstilles (automatisk), når man kommer ind I en ISR.
  Derfor bliver interrupts disabled.

- Det globale interruptflag bliver (automatisk) sat, når RETI udføres.
  Derfor bliver interrupt enabled igen.

- Hvis man vil tillade et interrupt i et andet interrupt, kan man – i starten af ISR’en – udføre sei().

Interrupts og resource konflikt

- Hvad kan gå galt i nedenstående?

```
.INCLUDE "M32DEF.INC"
.ORG        0x0         ;location for reset
            JMP         MAIN
.ORG        0x14        ;Timer0 compare match
            JMP         T0_CM_ISR
;---------main program-----------------------------
.ORG        0x100
MAIN:       LDI         R20,HIGH(RAMEND)
            OUT         SPH,R20
            LDI         R20,LOW(RAMEND)
            OUT         SPL,R20  ;set up stack
            SBI         DDRB,5   ;PB5 = output
            LDI         R20,160
            OUT         OCR0,R20
            LDI         R20,0x09
            OUT         TCCR0,R20
            LDI         R20,(1<<OCIE0)
            OUT         TIMSK,R20
            SEI
            LDI         R20,0xFF
            OUT         DDRC,R20
            OUT         DDRD,R20
            LDI         R20, 0
HERE:       OUT         PORTC,R20
            INC         R20
            JMP         HERE
;------------------------ISR for Timer0
T0_CM_ISR:
            IN          R20,PIND
            INC         R20
            OUT         PORTD,R20
            RETI
```

Løsning 1: Brug af forskellige registre

- Vælg nogen andre registre (bøvlet).

```
.INCLUDE "M32DEF.INC"
.ORG      0x0       ;location for reset
          JMP       MAIN
.ORG      0x14      ;Timer0 compare match
          JMP       T0_CM_ISR
;---------main program-----------------------------
.ORG      0x100
MAIN:     LDI       R20,HIGH(RAMEND)
          OUT       SPH,R20
          LDI       R20,LOW(RAMEND)
          OUT       SPL,R20  ;set up stack
          SBI       DDRB,5   ;PB5 = output
          LDI       R20,160
          OUT       OCR0,R20
          LDI       R20,0x09
          OUT       TCCR0,R20
          LDI       R20,(1<<OCIE0)
          OUT       TIMSK,R20
          SEI
          LDI       R20,0xFF
          OUT       DDRC,R20
          OUT       DDRD,R20
          LDI       R20, 0
HERE:     OUT       PORTC,R20
          INC       R20
          JMP       HERE
;-------------------------ISR for Timer0
T0_CM_ISR:
          IN        R21,PIND
          INC       R21
          OUT       PORTD,R21
          RETI
```

Løsning 2: “Context saving” (PUSH/POP)

- PUSH de anvendte registre i starten af ISR.
  POP dem tilbage igen inden IRET.

```
.INCLUDE "M32DEF.INC"
.ORG      0x0       ;location for reset
          JMP       MAIN
.ORG      0x14      ;Timer0 compare match
          JMP       T0_CM_ISR
;---------main program----------------------------
.ORG      0x100
MAIN:     LDI       R20,HIGH(RAMEND)
          OUT       SPH,R20
          LDI       R20,LOW(RAMEND)
          OUT       SPL,R20  ;set up stack
          SBI       DDRB,5   ;PB5 = output
          LDI       R20,160
          OUT       OCR0,R20
          LDI       R20,0x09
          OUT       TCCR0,R20
          LDI       R20,(1<<OCIE0)
          OUT       TIMSK,R20
          SEI
          LDI       R20,0xFF
          OUT       DDRC,R20
          OUT       DDRD,R20
          LDI       R20, 0
HERE:     OUT       PORTC,R20
          INC       R20
          JMP       HERE
;--------------------------ISR for Timer0
T0_CM_ISR:
          PUSH      R20      ;save R20
          IN        R20,PIND
          INC       R20
          OUT       PORTD,R20
          POP       R20      ;restore R20
          RETI
```

I assembly: Husk at gemme SREG

- Hvis assembly: PUSH/POP altid SREG I ISR, hvis flagene ændres i ISR.

```
PUSH R20
IN   R20,SREG
PUSH R20
...
POP  R20
OUT  SREG,R20
POP  R20
```

Slut på lektion 16

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING