Praktikum Mikrorechner 10 (Interrupt)

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal

5. November 2014

Interrupt Prinzip

Interrupt

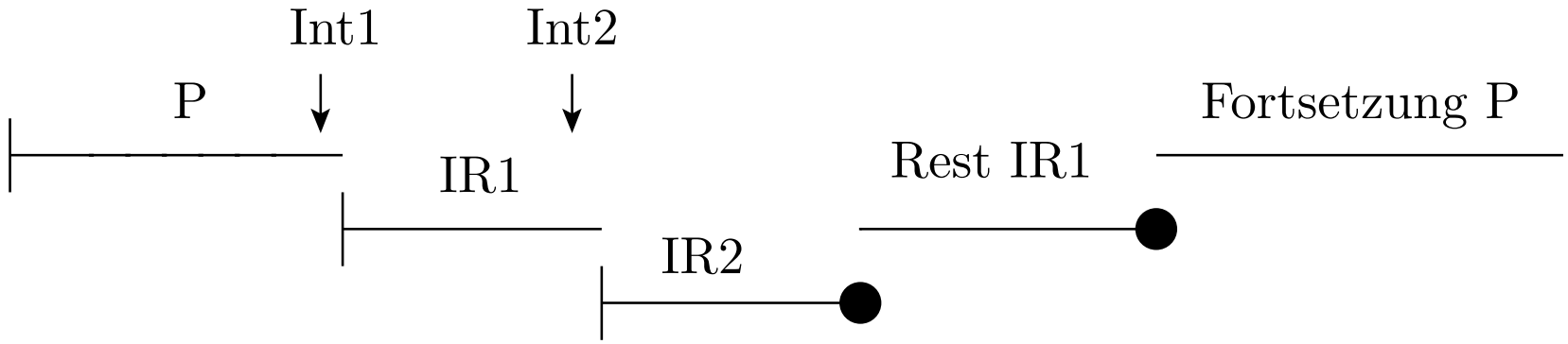
In der Praxis wartet ein Rechner sehr oft:

- für eine bestimmte Zeit (z.B. eine bestimmte Ausgabepulsbreite)
- auf ein Ereignis (z.B. eine Eingabe)

Dafür gibt es 2 Möglichkeiten:

- Warteschleife (Polling, der Rechner tut in der Zeit nichts anderes)

- Interrupt: Der Rechner bearbeitet ein anderes Programm oder befindet sich in einer Warteschleife. Bei einem eingehenden Ereignis:
  - Pegelwechsel am externen Interrupteingang
  - Zeichen über SIO empfangen / versendet
  - Zeitzählerüberlauf
  - ...

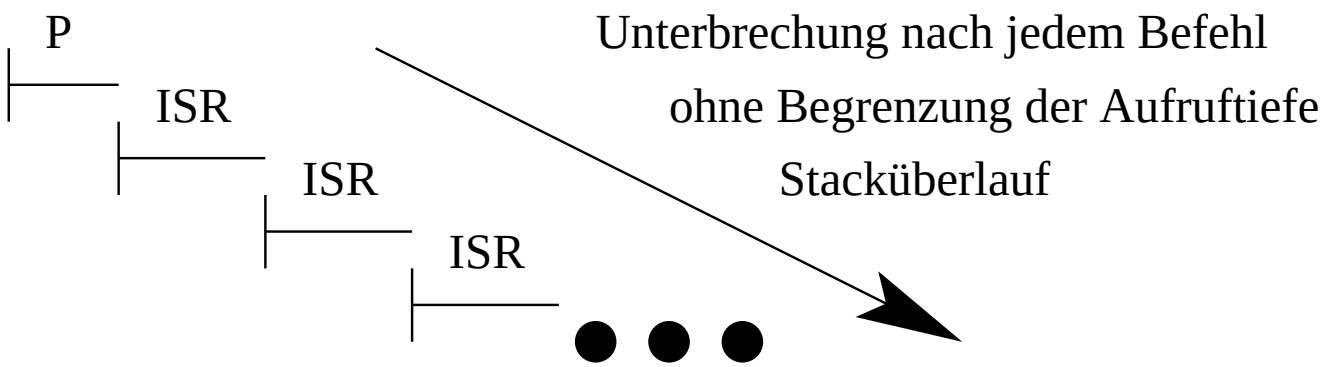


P	Programm	Int1	Ereignis, dass IR1 startet
IR1	Interruptroutine Prioriaet 1	Int2	Ereignis, dass IR2 startet
IR2	Interruptroutine Prioriaet 2		

Interruptquellen

Interruptquelle	Interruptflag(s)	Startadresse
externer Int. 0	IE0	0003h
Timer 0	TF0	000bh
externer Int. 1	IE1	0013h
Timer 1	TF1	001bh
SIO	RI, TI	0023h
Timer 2	TF2, EXF2	002bh
AD-Wandler	IADC	0043h
externer Int. 2	IE2	004bh
CAPCOM Emergency	TRF, BCERR	0053h
Compare Timer 2	CT2P	005bh
Cature/Compare Match	CCx	0063h
Compare Timer 1	CT1FP, CT1FC	006bh
Power Down		007bh

Unbeschränkte Unterbrechungserlaubnis



Beschränkung der Unterbrechungsmöglichkeiten

- zwei Interruptprioritäten, Unterbrechung nur bei höherer Priorität ⇒ max. Aufruftiefe 2

ip0 (adr. b8h)			PT2	PS	PT1	PX1	PT0	PX0
ip1 (adr. b9h)			PCT1	PCCM	PCT2	PCEM	PX2	PADC

PXn	externer Interrupt n
PTn	Timerinterrupt n
PS	serieller Interrupt
PADC	Interrupt des Analog-Digital-Wandler

- reti (return from interrupt): spezieller Rücksprungbefehl aus Interruptroutinen, der die »aktuelle Interruptebene« zurücksetzt

- Interruptfreigabe global oder für alle Quellen einzeln

ien0 (adr. a8h)	EA		ET2	ES	ET1	EX1	ET0	EX0
ie01 (adr. a9h)			ECT1	ECCM	ECT2	ECEM	EX2	EADC

EA	globale Freigabe
EXn	externer Interrupt n
ETn	Timer-Interrupt n
ES	serieller Interrupt
EADC	Interrupt des Analog-Digital-Wandler

Programm mit Interruptroutine

```
org 0   ; Start nach Power-on und Reset
  ljmp Start
...
org <isr-Startadresse>
  ljmp isr_xx

org <Startadresse>
Start:
  clr EA          ; alle Interrupts aus
  mov sp, #90h    ; Stack vorbereiten
  setb E<xx>      ; Interrupt xx aktivieren
  setb EA         ; Interrupts global erlauben
Wiederhole:
      Wenn Aktionsflag fuer Aufgabe 1 gesetzt,
           dann arbeite Aufgabe 1 ab
```

```
      Wenn Aktionsflag fuer Aufgabe 2 gesetzt,
            dann arbeite Aufgabe 2 ab
      ...
      ljmp Wiederhole
  ...
isr_xx:
  jnb <Interruptflag>, isr_xx_ex
  push acc
  push psw
   <kleine Aufgabe, meist Daten kopieren>
    setb <Aktionsflag isr-xx>
  pop psw
  pop acc
isr_xx_ex:
  reti
```

Externer Interrupt

Auslösendes Ereignis: P3.2 (Int0), P3.3 (Int1), P3.6 (Int2)

- Low
- fallende Flanke
- steigende Flanke
- beide Flanken

itcon (adr. 9ah)

IT2	IE2	I2ETF	I2ETR	I1ETF	I1ETR	I0ETF	I0ETR

ITx=0 Interrupt bei Low

ITx=1 Interrrupt Signalflanke

IxETF=1 bei fallender Flanke

IxETR=1 bei steigender Flanke

Experiment 1

Zählen der Flanken am Interrupteingang P3.2 (Int0) mit einer Interruptroutine und Zählstandausgabe an Port 1

```
    org 3   ; externer Interrupt 0
      inc P1
            ; clr IE0 Anforderungsflag löschen;
            ; hier nicht erforderlich)
     reti
```

```
;-------------------------------
; Hauptprogramm
org 100h
  clr  EA          ; alle Interrupts aus
  mov  SP, #0c0h   ; Stack einrichten
  mov  P1, #0      ; Anfangswert für P1
  setb P3.2        ; als Eingang vorbereiten
  setb EX0         ; Freigabe des ext. Int.0
  clr  PX0         ; niedrige Priorität
  setb IT0         ; Interrupt bei Flanke
  mov  ITCON, #3   ; beide Flanken
  setb EA          ; Interrupt globale ein
  ljmp $           ; Endlosschleife
```

Experiment 2

Interrupt bei P3.2=0 (Int0, pegelgesteuert). isr soll nur 10µs Rechenzeit verbrauchen und wird bei gedrückter Taste nach jedem Befehl des Hauptprogramms eingefügt. Das Hauptprogramm soll ohne isr-Aufrufe ein 2,5Hz Blinksignal erzeugen.

```
org 3                ; externer Interrupt 0
ljmp Int_EX0
 ...
; Hauptprogramm
org 100h
  clr EA             ; alle Interrupts verbieten
  mov SP, #0c0h      ; Stack einrichten
  mov P1, #0         ; Anfangswert für P1
  setb P3.2          ; Interrupteingang vorbereiten
  setb EX0           ; Freigabe ext. Int.0
  clr  PX0           ; niedrige Priorität
  clr  IT0           ; Interrupt bei P3.2=0
  setb EA            ; Interrupt global ein
```

```
loop:
  wait_nx100ms_mac 2
  cpl P1.0
  ljmp loop
;================================

Int_EX0:
  nop
  ... Insg. 18 nop-Befehle; incl. reti 20
      Zyklen bei 12 MHz Prozessortakt
      => 10µs
  reti
```

Was passiert, wenn Taste P3.2 gedrückt wird?

Aufgaben

Aufgabe 10.1: Experimente mit dem externen Interrupt

- Führen sie die beiden Experimente mit dem externen Interrupt durch.