

Interrupt beim AIM-65

IRQ und NMI beim AIM 65

Der Verfasser erstand einen AIM 65 in der Absicht, sich einmal ohne den Umweg über eine höhere Programmiersprache mit der Programmierung des 6502 auseinanderzusetzen und vielleicht auch einige selbstgelötete Externa anzusteuern. Der AIM 65 scheint alle diesbezüglichen Erwartungen in sehr hohem Maße zu erfüllen und die Serie "Lernen mit AIM 65" von Manuel Krüger bot eine willkommene Hilfestellung. Leider wird sie nicht fortgesetzt. Es wird hier nun nicht versucht, eine Nachfolge anzutreten. Wohl aber könnte der folgende Beitrag dem, der in der Frage der Interruptbehandlung noch nicht soweit ist, einen Schritt weiterhelfen. In gleicher Weise würde sich der Verfasser freuen, wenn vielleicht durch Ergänzungen oder Folgebeiträge auch ihm weitergeholfen würde.

Und nun zum Thema:

Über die beiden Leitungen $\overline{\text{NMI}}$ (non-maskable interrupt, Pin 6 an J3) und $\overline{\text{IRQ}}$ (interrupt request, Pin 4) kann der AIM Impulse aufnehmen, die a) eine Unterbrechung des laufenden Programms bewirken und b) stattdessen Interruptprogramme ablaufen lassen, nach denen in das ursprüngliche Programm zurückgekehrt werden kann. Dabei kann der IRQ maskiert werden: Ist im Prozessorstatus das I-Flag gesetzt (SEI-Befehl oder akuter Interruptablauf), so wird ein (erneuter) IRQ nicht angenommen. Zur Abnahme eines IRQ muß das Interrupt-Flag gelöscht sein (CLI oder RTI). Der NMI ist nicht in dieser Weise maskierbar; er wird immer angenommen.

Während der $\overline{\text{IRQ}}$ schon auf einfachen Masseanschluß (Taster oder Zusammendrücken zweier Drähte) reagiert, erfordert der $\overline{\text{NMI}}$ eine besondere Impulsform, eine fallende Flanke. Eine solche erzeugt man am einfachsten mit einem R-S-Flipflop, das man aus zwei NAND-Gattern zusammenschalten kann. Da ein SN 7400 vier NAND-Gatter enthält, bedient der Verfasser auch den $\overline{\text{IRQ}}$ auf diesem Wege. Die benötigte Versorgungsspannung wird an J3 entnommen (Pin 22 ist Masse, Pin 21 führt +5V). Zum Nachbau mögen folgende Skizzen genügen:

Das folgende Programm hat nur den einen Sinn, die vorgetragenen Überlegungen zu konkretisieren; es ist ein Demo: Das Hauptprogramm gibt ständig "PROGRM" über den Drucker aus, das IRQ-Programm gibt auf dem gleichen Wege "IRQ" aus und das NMI-Programm "NMI". Wird bei laufendem Hauptprogramm einer der beiden Interrupttaster (Umschalter) betätigt, so wird der Ausdruck des Wortes "PROGRAM", je nachdem, an welcher Stelle des Programmablaufs der Interrupt erfolgte, unterbrochen, das jeweilige andere Kennwort wird ausgegeben und anschließend die Ausgabe von "PROGRAM" fortgesetzt, sofern der Interrupt nicht den Ausgabebufferinhalt überschrieb. Es kann zu Schmierdruck kommen, auch zu der Meldung "PRINTER DOWN", wonach der Drucker wieder eingeschaltet werden kann.

Unser Interrupt-Demo beginnt in 0200, das IRQ-Programm in 0226 und das NMI-Programm in 023F. 0258 ist eine Verzögerungsroutine und 0260...026F eine Tabelle der benötigten ASCII-Codes.

Ab 0200 wird zunächst die Anfangsadresse des IRQ-Programms nach A404 (low) und A405 (high) gespeichert und die des NMI-Programms entsprechend nach A402 und A403.

Ab 0215 werden mit indizierter Adressierung die ASCII-Codes für das Kennwort "PROGRAM" aus der Tabelle geholt und gedruckt. (0D ist carriage return/line feed). Dieser Programmteil bildet in sich eine endlose Schleife (JMP 0215), sofern nicht durch einen Interrupt unterbrochen. Im IRQ-Programm werden zunächst die Inhalte von Akku, X- und Y-Register gerettet. (Der Prozessorstatus wird automatisch beim Eingang des Interrupt-Impulses gerettet). Anschließend wird wieder mit indizierter Adressierung und mittels der Tabelle das Wort "IRQ" ausgegeben. Abschließend werden – in umgekehrter Reihenfolge – die Y-, X- und A-Werte vom Stack zurückgeholt. Der Prozessorstatus wird durch das RTI automatisch unter Löschung des Interrupt-Flags wieder hergestellt.

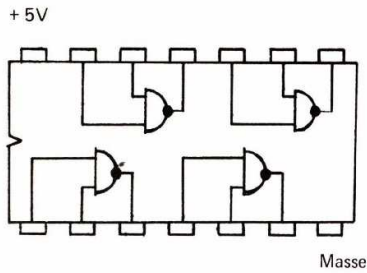


Bild 1: Anschlußbelegung des SN 7400 (v. oben)

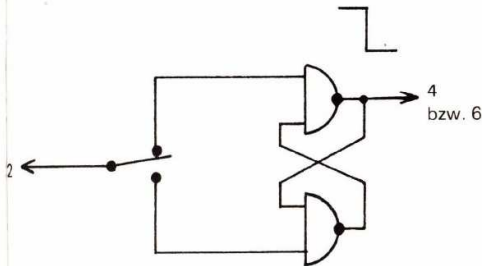


Bild 2: R-S-Flipflop aus 2/4 SN 7400

Das NMI-Programm (ab 023F) bewirkt das Entsprechende. Als praktische Anwendungen erscheinen z.B. möglich:

- Es läuft ein beliebiges Hauptprogramm und gleichzeitig wird ein externer Steuerungsvorgang überwacht. Löst dieser einen Interrupt aus, so übernimmt AIM den Meßwert, errechnet die erforderliche Reaktion, gibt die entsprechenden Werte aus und kehrt in das Hauptprogramm zurück.
- Man könnte sich ein Spiel vorstellen, in dessen Verlauf der AIM "knobelt" (Pseudo-Random-Programm); wird eine Interrupttaste betätigt, so wird das Knobeln unterbrochen und sein Ergebnis angezeigt. Anschließend Rückkehr ins Hauptprogramm mit der Alternative "Knobelerggebnis akzeptiert" oder "Weiterknobeln!".

ROCKWELL AIM 65

<1>

```
0000      *=0200
0200  A9  LDA  # 26
0202  8D  STA  A404
0205  A9  LDA  # 02
0207  8D  STA  A405
```

```
020A  58  CLI
020B  A9  LDA  # 3F
020D  8D  STA  A402
0210  A9  LDA  # 02
0212  8D  STA  A403

0215  A2  LDX  # 08
0217  BD  LDA  0267,X
021A  20  JSR  E9BC
021D  CA  DEX
021E  D0  BNE  0217
0220  20  JSR  0258
0223  4C  JMP  0215

0226  48  PHA
0227  8A  TXA
0228  48  PHA
0229  98  TYA
022A  48  PHA
022B  A2  LDX  # 04
022D  BD  LDA  0263,X
0230  20  JSR  E9BC
0233  CA  DEX
0234  D0  BNE  022D
0236  20  JSR  0258
0239  68  PLA
023A  A8  TAY
023B  68  PLA
023C  AA  TAX
023D  68  PLA
023E  40  RTI

023F  48  PHA
0240  8A  TXA
0241  48  PHA
0242  98  TYA
0243  48  PHA
0244  A2  LDX  # 04
0246  BD  LDA  025F,X
0249  20  JSR  E9BC
024C  CA  DEX
024D  D0  BNE  0246
024F  20  JSR  0258
0252  68  PLA
0253  A8  TAY
0254  68  PLA
0255  AA  TAX
0256  68  PLA
0257  40  RTI
0258  CA  DEX
0259  88  DEY
025A  D0  BNE  0259
025C  8A  TXA
025D  D0  BNE  0258
025F  60  RTS

0260
```

