

Kapitel 18

Externe Komponenten

Anschluss von externen Komponenten

- Einfachste Art : Direkt an einem Port
 - Beispiel Ausgabe : 7-Strich Nummerndisplay
 - Beispiel Eingabe : Schalterblock (Mäuseklavier)
 - 8051 Zugriff mit Befehlen zum internen Datenspeicheradressraum

 Eventuell Verschwendung einer wertvollen Resource

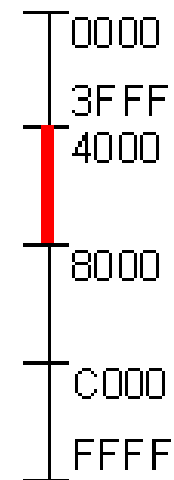
- Typische Anwendungsart : Am externen Systembus
 - 8051 Zugriff mit MOVX Befehl

Anschluss am externen Systembus

Externe Eingabe/Ausgabe-Bausteine können in der gleichen Weise wie externe Speicherbausteine an den externen Systembus angeschlossen werden.

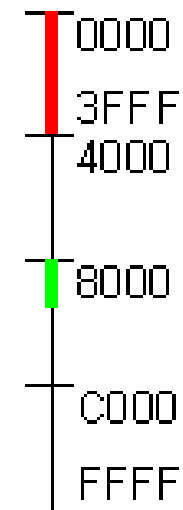
● Adressierung

- Der Adressdekoder überprüft die höherwertigen Bits des Adressbusses (A15 ... An+1) und aktiviert das Chip Select Signal (\overline{CS}), wenn die Adresse dem externen Speicherbaustein oder dem externen E/A-Baustein zugeordnet ist.
- Die übrigen Bits auf dem Adressbus (An ... A0) identifizieren ein Element **innerhalb** des selektierten externen Bausteins.
- **Beispiel** : 16 kB Speicherbaustein bei Adresse 0x4000 anschließen.
 - Gültige Adresse : 01xx xxxx xxxx xxxx (binär)
 - 16 kB = 2^{14} Bytes → A13 ... A0 für die Adressierung innerhalb des Bausteins.
 - Dekoder überprüft A15 ... A14.
 - Wenn A15,A14 = 0,1 wird \overline{CS} aktiviert.
 - Ein Dekoder innerhalb des externen Speicherbausteins benutzt die niederwertigen 14 Bits (A13 ... A0), um das gewünschte Byte auszuwählen.

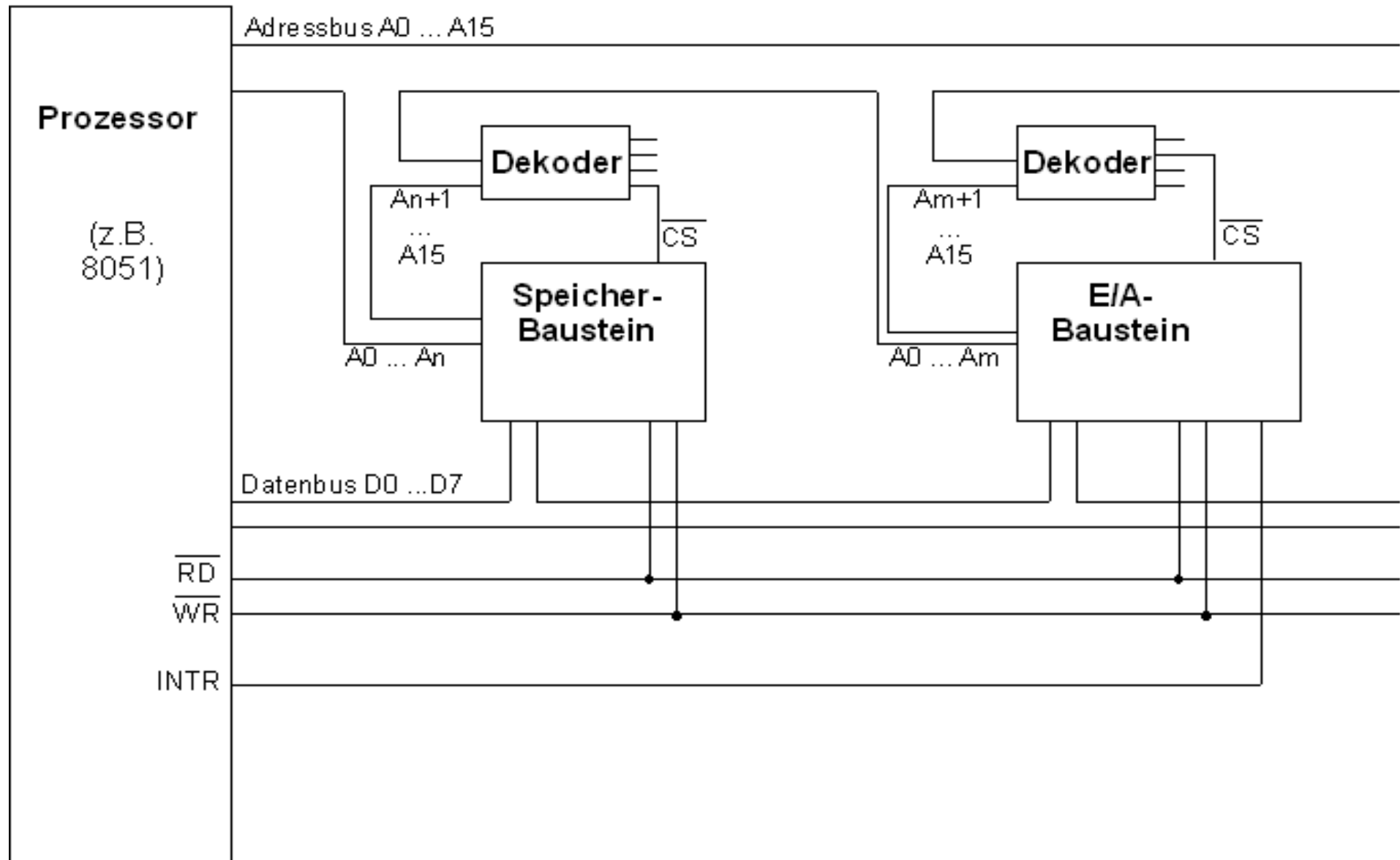


Memory-mapped-I/O

- Die Adressbereiche für die Datenspeicherbausteine und die Adressbereiche für die E/A-Bausteine befinden sich **in einem einzigen Adressraum**.
 - Die Adressbereiche dieser externen Bausteine dürfen sich deshalb nicht überlappen. Sonst gibt es eine Doppeladressierung.
 - Externe Speicher- und E/A-Bausteine werden mit den selben Befehlen angesprochen.
 - Der 8051 Prozessor verwendet die **Memory-mapped-I/O**-Methode. Für Datenzugriffe benutzt der 8051 Prozessor den Befehl **MOVX**.
- **Beispiel** : E/A-Baustein mit 4 Byte-Registern an Adressen **0x8000 ... 0x8003** und ein 16 kB Speicherbaustein bei **0x0000**.
 - A15 ... A0 für die Adressierung der 4 Register innerhalb des E/A-Bausteins.
 - Dekoder überprüft A15 ... A2 im Fall einer **vollständigen Dekodierung**.
 - Manchmal ist auch **unvollständige Dekodierung** ausreichend.
 - Bei unvollständiger Dekodierung werden nicht alle zu dekodierenden Adressbits überprüft.
 - Konsequenz ist, dass ein Element über mehrere Adressen angesprochen werden kann.
 - Wenn im vorliegenden Beispiel im externen Dekoder nur A15 = 1 überprüft wird, dann ist das Byte bei 0x8000 auch über 0x8004, 0x8008, usw. erreichbar.
 - Solange kein anderer Baustein oberhalb von 0x8000 angeschlossen werden soll, ist das eine mögliche Lösung.
 - Eine Adresse mit A15 = 0 selektiert den Speicherbaustein.



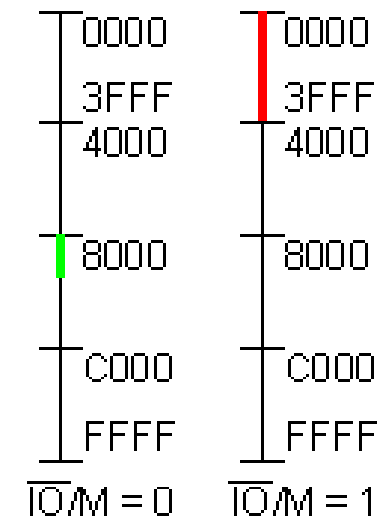
Anschluss externer Bausteine



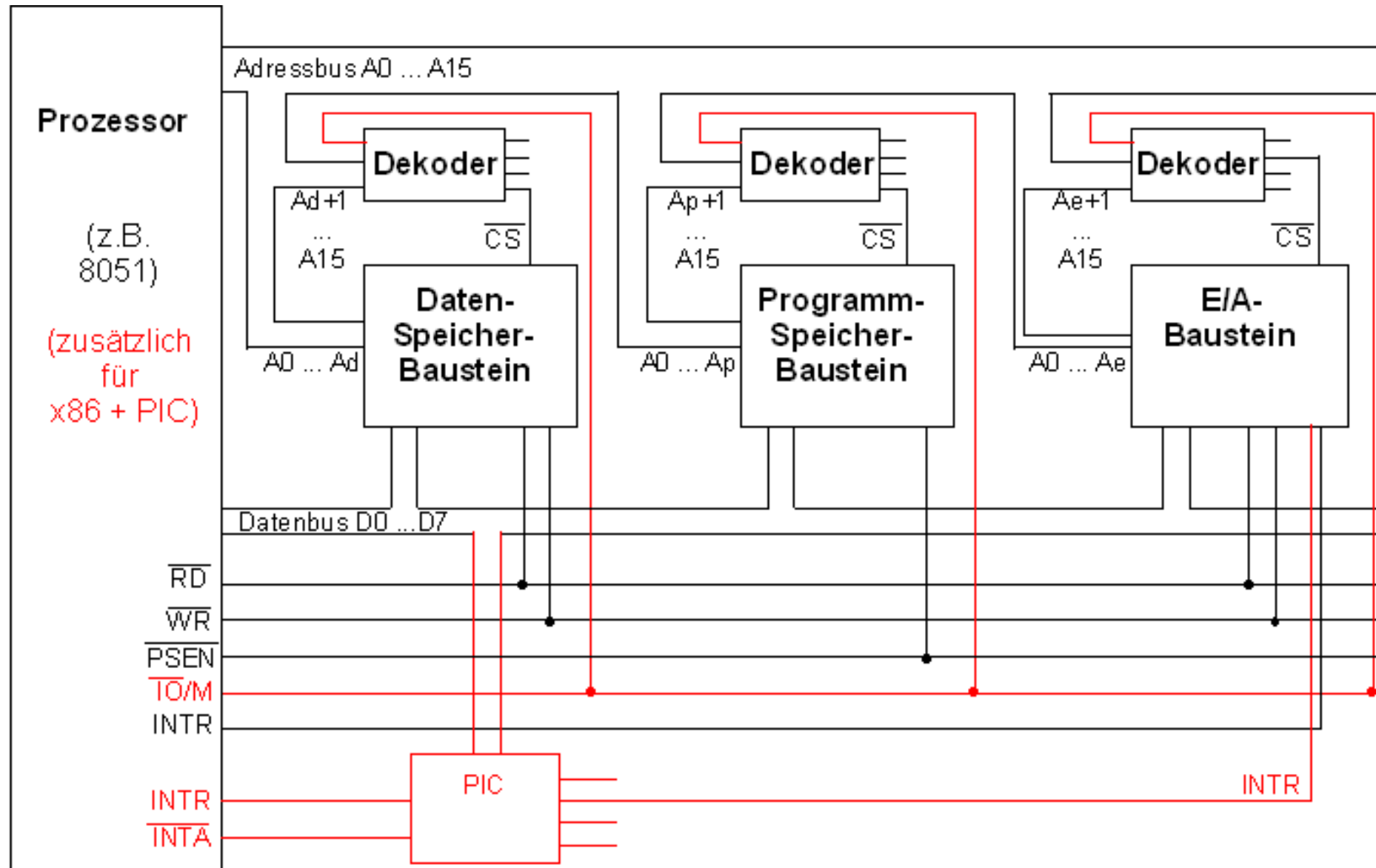
vollständige Adressierung !

Isolated-I/O

- **zwei Adressräume**, ein Adressraum für Speicherbausteine und ein anderer Adressraum für E/A-Bausteine.
- Die zusätzliche Steuerleitung $\overline{IO/M}$ unterscheidet, welcher der beiden Adressräume angesprochen ist :
 $\overline{IO/M} = 0$: Der I/O-Adressraum ist angesprochen.
 $\overline{IO/M} = 1$: Der Speicher-Adressraum ist angesprochen.
- Die x86 Prozessorfamilie verwendet die Isolated-I/O-Methode.
- Für Datenzugriffe zum **Speicheradressraum** benutzt der x86 Prozessor die **MOV** Befehle.
- Für Datenzugriffe zum **I/O-Adressraum** benutzt der x86 Prozessor die Befehle **IN** und **OUT**. Sie übertragen ein Byte zwischen dem Akkumulator und der E/A-Einheit.
- Für **IN** und **OUT** Befehle setzt der Prozessor die $\overline{IO/M}$ Steuerleitung auf 0.
- Für **MOV** Befehle setzt der Prozessor die $\overline{IO/M}$ Steuerleitung auf 1.
- Die Adressdekoer berücksichtigen zusätzlich die $\overline{IO/M}$ Steuerleitung für die Erzeugung des \overline{CS} Signals.
- Ein Speicherbaustein kann durchaus den gleichen Adressbereich benutzen wie ein E/A-Baustein. Die $\overline{IO/M}$ Steuerleitung unterscheidet zwischen den beiden Bereichen.
- Eine unvollständige Adressdekodierung ist bei Isolated I/O meist einfacher, da nur E/A-Adressbereiche gegen einander abgegrenzt werden müssen.
- Für das Lesen und Schreiben von Elementen der externen Bausteine (Speicher und E/A) aktiviert der Prozessor die \overline{RD} bzw. die \overline{WR} Steuerleitung.
- Das zu lesende oder zu schreibende Datenbyte wird über den externen Datenbus zwischen dem Prozessor und dem externen Baustein übertragen.



Anschluss externer Bausteine

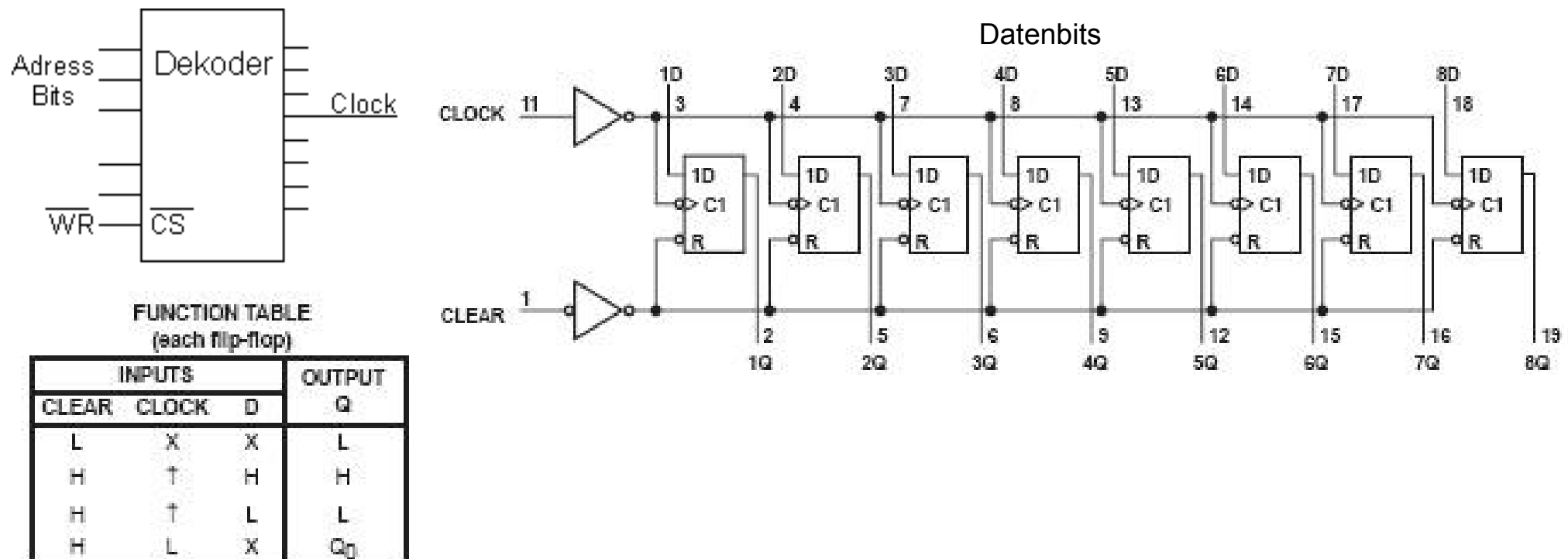


Interrupts

- Speicherbausteine benutzen keine Interrupts.
- Manche E/A-Bausteine benutzen Interrupts, andere (meist einfache) E/A-Bausteine benutzen keine Interrupts.
- Bei Systemen mit dem 8051 Prozessor werden die Interruptleitungen der E/A-Bausteine direkt mit den External Interrupt Anschlüssen des 8051 Prozessors verbunden.
- Bei Systemen mit x86 Prozessoren übernimmt der Peripheral Interrupt Controller (PIC) die Steuerung der Interrupts.
 - Die externen E/A-Bausteine haben individuelle Interruptleitungen zum PIC.
 - Bei Vorliegen eines Interrupt Requests von einem externen E/A-Baustein aktiviert der PIC die INTR Steuerleitung zum Prozessor.
 - Wenn der Prozessor den Interrupt annehmen will, aktiviert er die \overline{INTA} Steuerleitung, worauf der PIC die Identifikationsnummer für den Interrupt auf den Datenbus stellt.
 - Auf Grund der Identifikationsnummer führt dann der Prozessor die entsprechende Interrupt Service Routine aus.

Einfache E/A-Bausteine

- Als **Output**-Schnittstelle kann z.B ein **74LS273** Portbaustein eingesetzt werden.
 - Entspricht in etwa der Funktionalität eines 8051 Ports (nur Output).
 - Zur Adressdekodierung werden Dekoderbausteine (z.B. 74LS138) verwendet.
 - Eventuell genügt auch eine unvollständige Dekodierung
 - Die \overline{WR} Steuerleitung könnte an die \overline{CS} Leitung des Dekoders angeschlossen werden.
 - Die ausgewählte Dekoderausgangsleitung wird mit der CLOCK Leitung des 74LS273 Portbausteins verbunden.
 - Ein MOVX Befehl überträgt das Byte vom Akkumulator in die 74LS273 Portlatches.

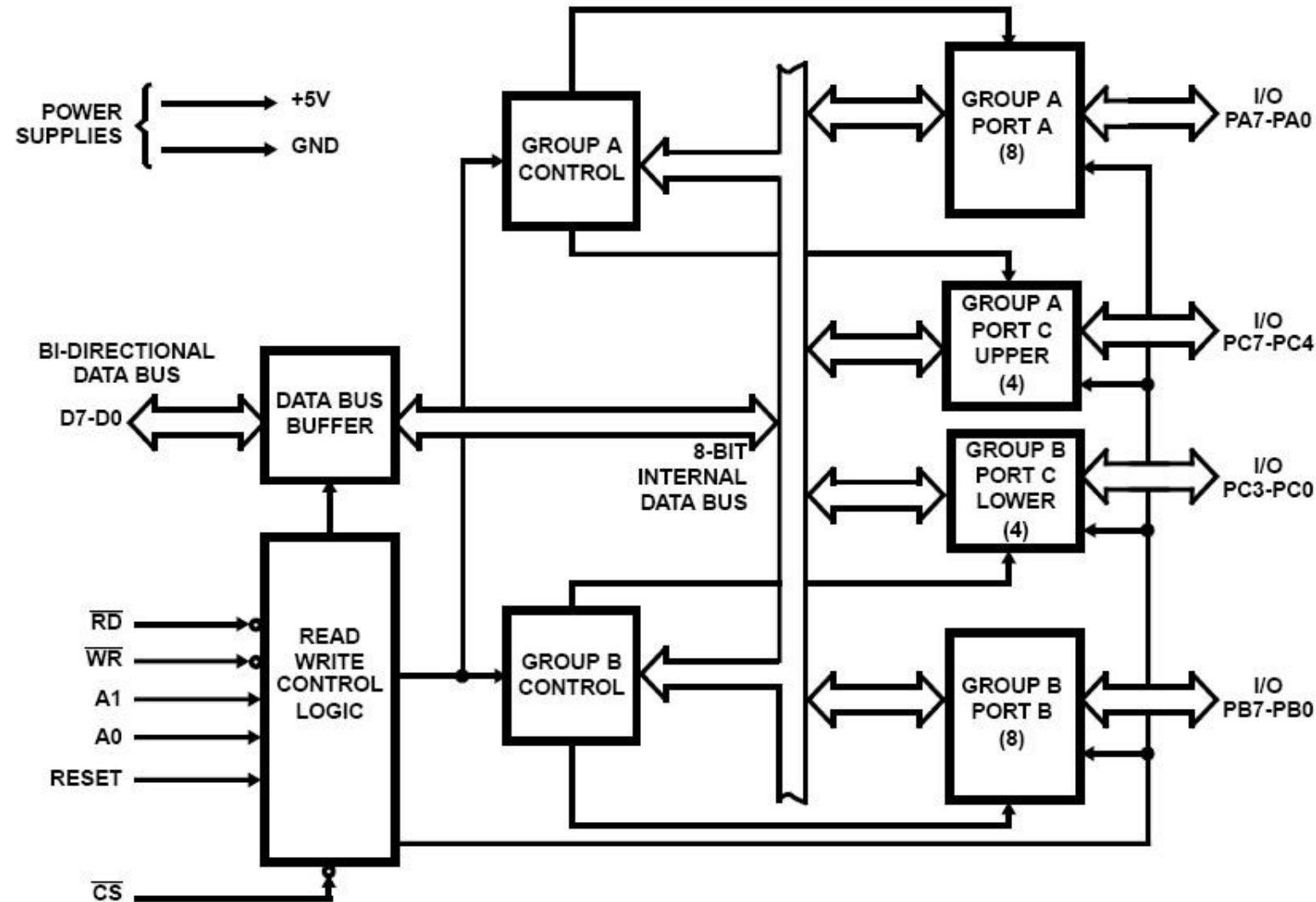


- Eine dem 8051 Port vergleichbare **Input**-Schnittstelle erfordert einfach 8 Tri-State-Treiber.
 - Ein MOVX Befehl wird benutzt, um das Input-Byte von der externen Schnittstelle einzulesen.
 - Eine richtige Adresse am Dekoder aktiviert die Dekoderausgangsleitung.
 - Die Dekoderausgangsleitung steuert die Enable Eingänge der Tri-State-Treiber.
 - Die externe Schaltung muss dafür sorgen, dass der Inputwert an den Eingängen der Tri-State-Treiber ansteht.

- Diese einfachen externen Anschlussbausteine benutzen keine Interrupts.

Schnittstellenbaustein 8255

- Zum Anschluss von vielen verschiedenartigen externen Komponenten und Geräten geeignet.
- Vielseitig konfigurierbar.
- Siehe auch <http://pdf1.alldatasheet.com/datasheet-pdf/view/66100/INTEL/8255A.html>



Schnittstellen des Bausteins

- Anschluss an den Prozessor über den externen Systembus
 - Adressbits A15 ... A2 (ggf. zusätzlich \overline{IO}/M) werden zur Dekodierung verwendet.
 - Adressbits A1, A0 können vier ein Byte große interne 8255 Elemente adressieren.
- Zur Peripherie hin gibt es drei Ports mit je 8 Bits : Port A, Port B, Port C.
- Einfache Betriebsart : Mode 0
 - Durchreichen von Bytes durch Port A, B und je 4 Bits durch Port C upper und Port C lower.
 - Jeder Port kann als Input oder Output konfiguriert werden.
- Komplexere Betriebsarten : Mode 1 und Mode 2
 - Zwei **Kanäle** (Dokumentation : Groups) werden gebildet : Kanal A und Kanal B
 - Kanal A benutzt Port A zur Datenübertragung und einige Bits von Port C zur Steuerung.
 - Kanal B benutzt Port B zur Datenübertragung und die übrigen Bits von Port C zur Steuerung.
 - Kanal A und Kanal B können unabhängig voneinander in Input- oder Output-Richtung benutzt werden
 - Die Betriebsart (Mode) von Kanal A und die Betriebsart von Kanal B bestimmen,
 - welche Port C Bits Kanal A bzw. Kanal B zugeordnet werden,
 - In Mode 1 und 2 sind Port C Bits 7 ... 3 Kanal A, und Port C Bits 2 ... 0 Kanal B zugeordnet.

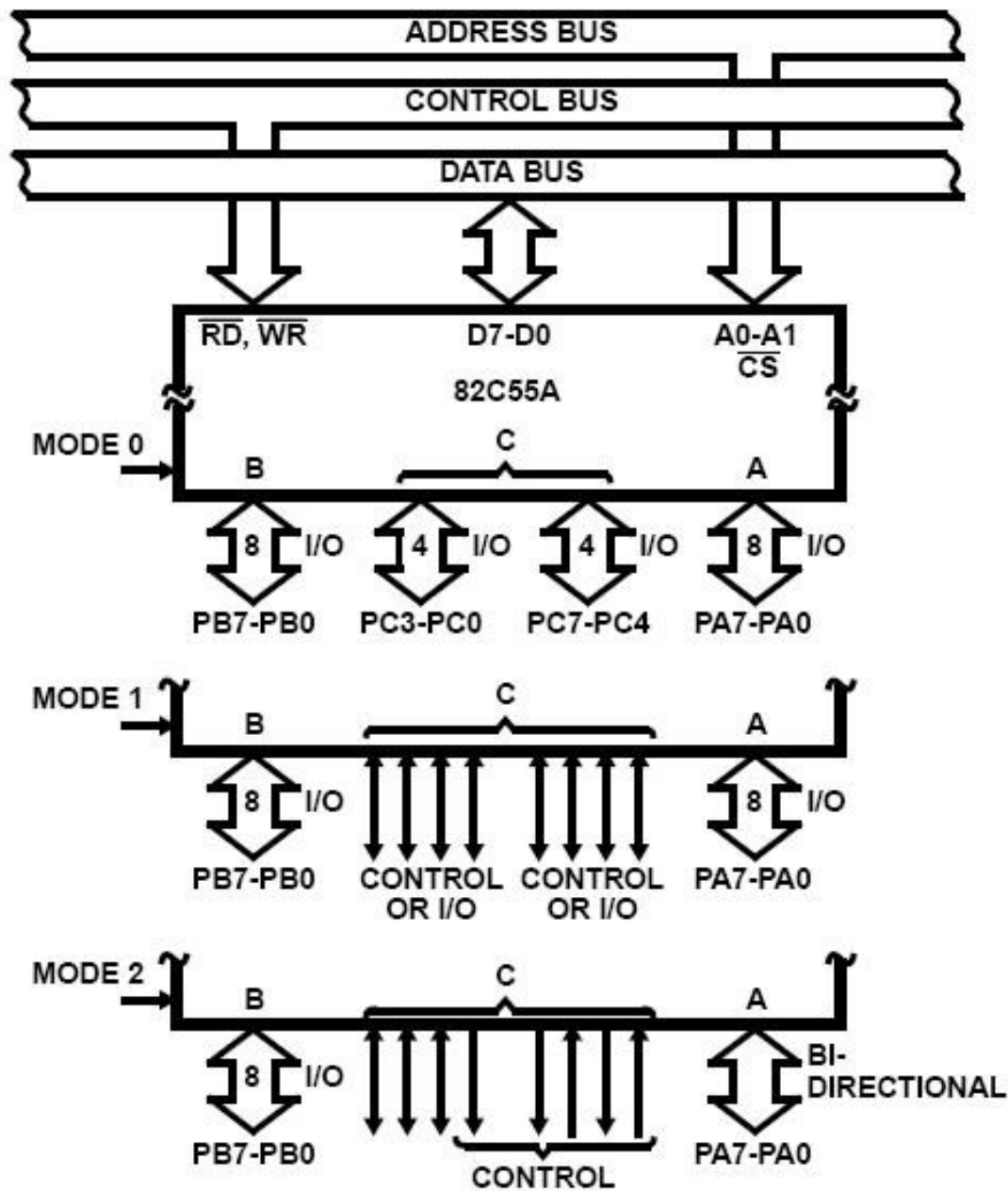


FIGURE 3. BASIC MODE DEFINITIONS AND BUS INTERFACE

Schnittstelle zwischen Prozessor und 8255

- Der Prozessor führt Lese- oder Schreiboperation aus
 - mit Hilfe von MOVX beim 8051 Prozessor, und IN oder OUT beim x86 Prozessor.
 - zu einer der vier 8255 Elemente, die durch A1 ... A0 adressiert werden.
 - Die Adressbits A15 ... A2 werden dekodiert und identifizieren den 8255 Baustein selbst.
 - Beispiel : Lies Port A

```
MOV          DPTR,#8000H ;Port A Adresse
MOVX        A,@DPTR    ;Lies den Wert von Port A (siehe nächste Seite)
```

- Der 8255-Baustein unterstützt die Benutzung von Interruptleitungen. Sie müssen aber nicht verwendet werden.

MOV Befehle vom/zum 8255

82C55A BASIC OPERATION

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
OUTPUT OPERATION (WRITE)					
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
DISABLE FUNCTION					
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

MOV Befehle vom/zum 8255 (Forts.)

- Eine **Schreiboperation zum Control Word** (mit Adressbits $A1, A0 = 1, 1$) führt eine von zwei Operationen aus, abhängig vom Bit 7 des geschriebenen Datenbytes.
 - Bit 7 = 1 : Die **Modes** für die beiden Kanäle und die **Richtung der Datenübertragungen** durch die Ports werden festgelegt.
 - Bit 7 = 0 : Die Bits von Port C werden bitweise gesetzt oder zurückgesetzt.
- Eine Leseoperation von der Adresse mit $A1, A0 = 1, 1$ gibt die aktuellen Modes für die Kanäle und die Richtungen der Datenübertragungen durch die Ports zurück.
 - Bei manchen 8255 Modellen wird diese Leseoperation nicht unterstützt.
- Beispiel : Setze
 - Kanal A auf Mode 0, Port A = Output, Port C (Bits 7 ... 4) = Input
 - Kanal B auf Mode 0, Port B = Input, Port C (Bits 3 ... 0) = Input

```
MOV    DPTR,#0x8003H    ;Control Byte Adresse
MOV    A,# 1 00 0 1 0 1 1 B ;Mode Spezifikation
MOVX   @DPTR,A         ;Setze den Mode
```

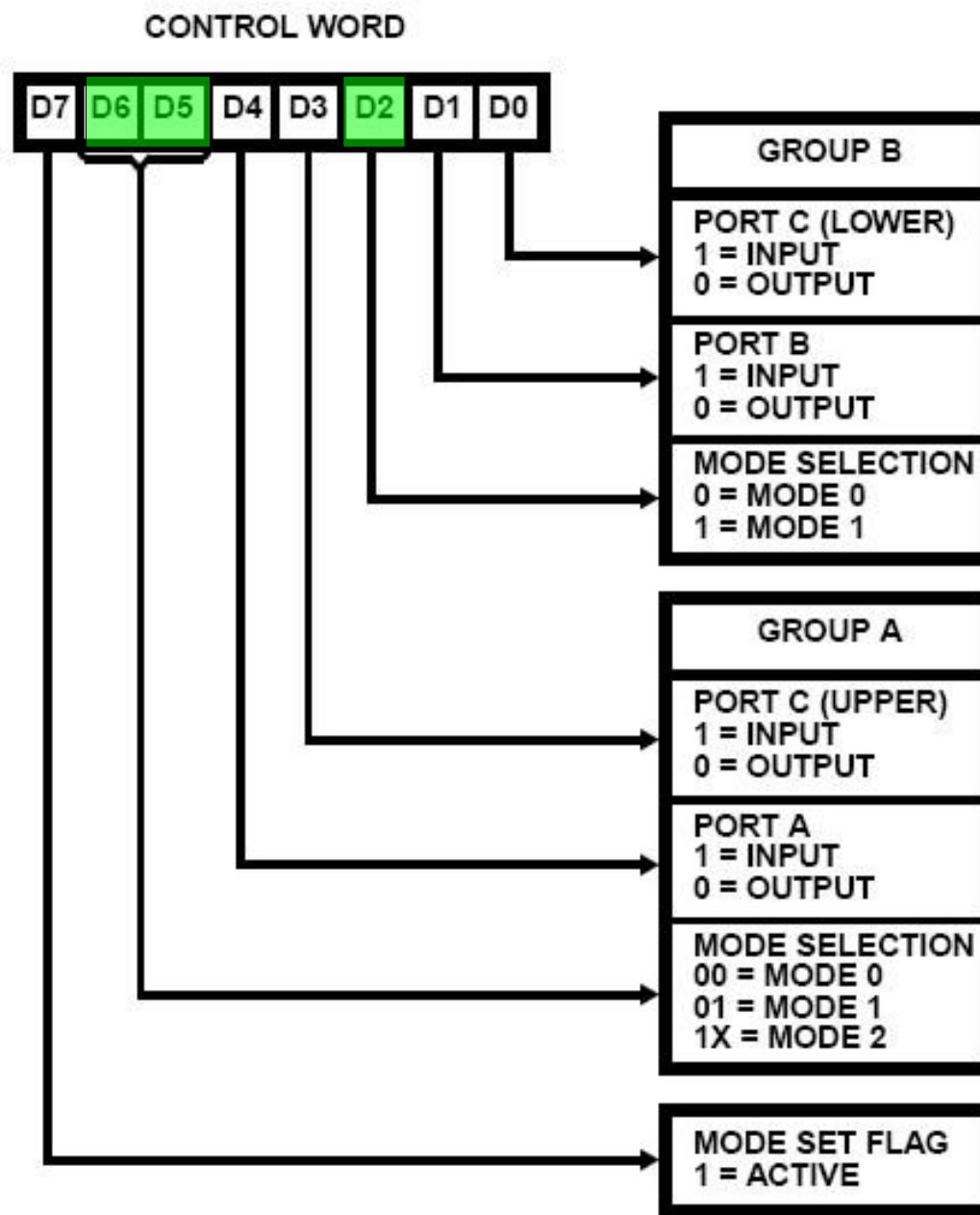



FIGURE 4. MODE DEFINITION FORMAT

Port C Bit Set/Reset (in Mode 1 und 2)

- Write zur Adresse mit A1,A0 = 1,1
- Datenbyte Bit 7 = 0

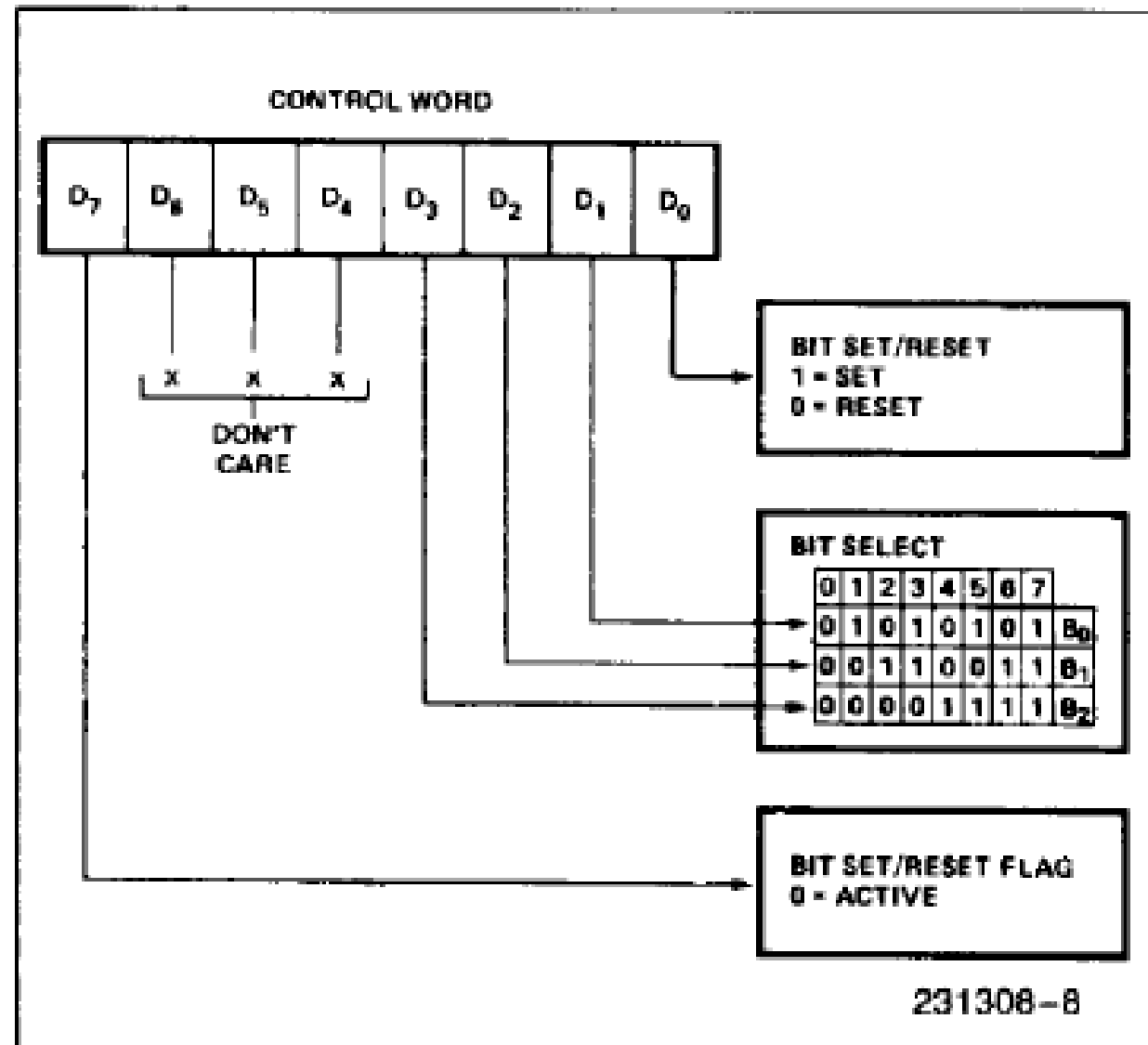


Figure 7. Bit Set/Reset Format

Mode 0 (für Kanal A und Kanal B)

- Zwei 8 Bit Schnittstellen und zwei 4 Bit Schnittstellen zum einfachen Durchreichen von Bits.
 - Kanal A : Port A, Port C Upper (Bits 7 ... 4)
 - Kanal B : Port B, Port C Lower (Bits 3 ... 0)
 - Jeder Port kann als Input oder als Output konfiguriert werden.
 - Outputdaten werden im 8255 Baustein verlatched, Inputdaten nicht (wie bei 8051 Ports).

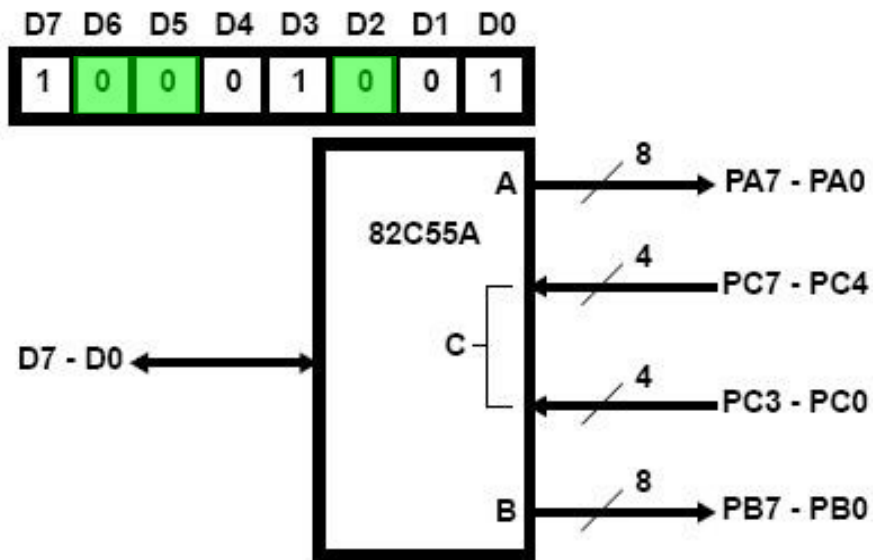
- Bei einer Schreiboperation nach Port C sind nur die Datenbits relevant, für die die Port C Bits als Output konfiguriert sind.
- Bei einer Leseoperation von Port C stammen nur die Datenbits von der Peripherie, für die die Port C Bits als Input konfiguriert sind.
- Beispielanwendungen :
 - Mode 0 - Output : 7-Strich Display zur Anzeige einer Ziffer.

```
MOV    DPTR,#0x8001 ;Port B Adresse
MOV    A,#0x54      ;Anzeigewert setzen
MOVX   @DPTR,A      ;Ziffer anzeigen
```

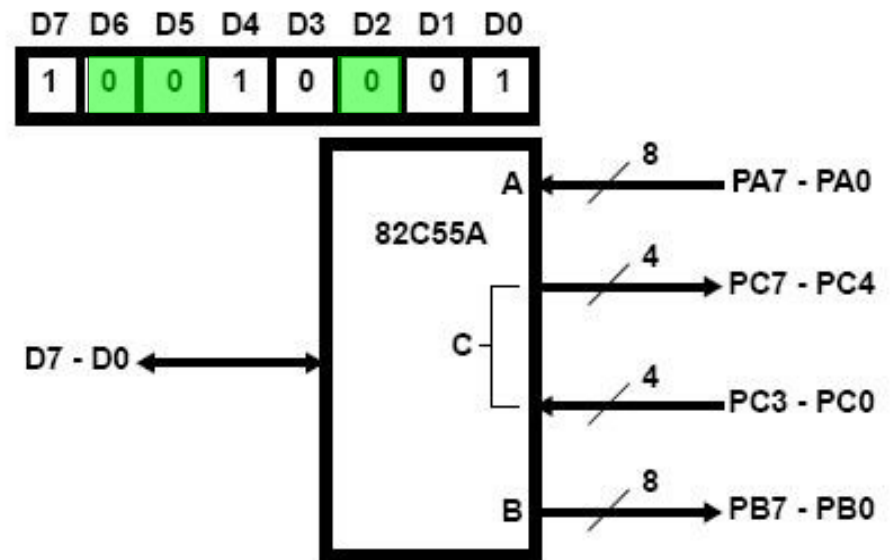
 - Mode 0 - Input : Schalterblock (Mäuseklavier) auslesen.

Mode 0

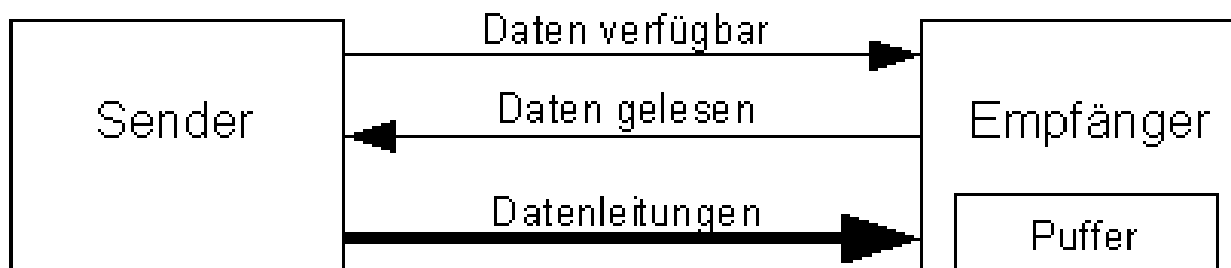
CONTROL WORD #5



CONTROL WORD #9



Handshake-Verfahren

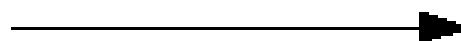


0. Daten verfügbar AUS

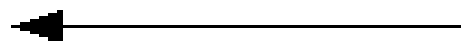
0. Daten gelesen AUS

1. Daten auf Datenleitungen

2. Daten verfügbar EIN

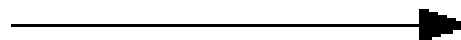


3. Daten in Puffer einlesen

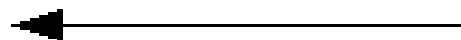


4. Daten gelesen EIN

5. Daten verfügbar AUS



6. Daten verarbeiten (Puffer leeren)



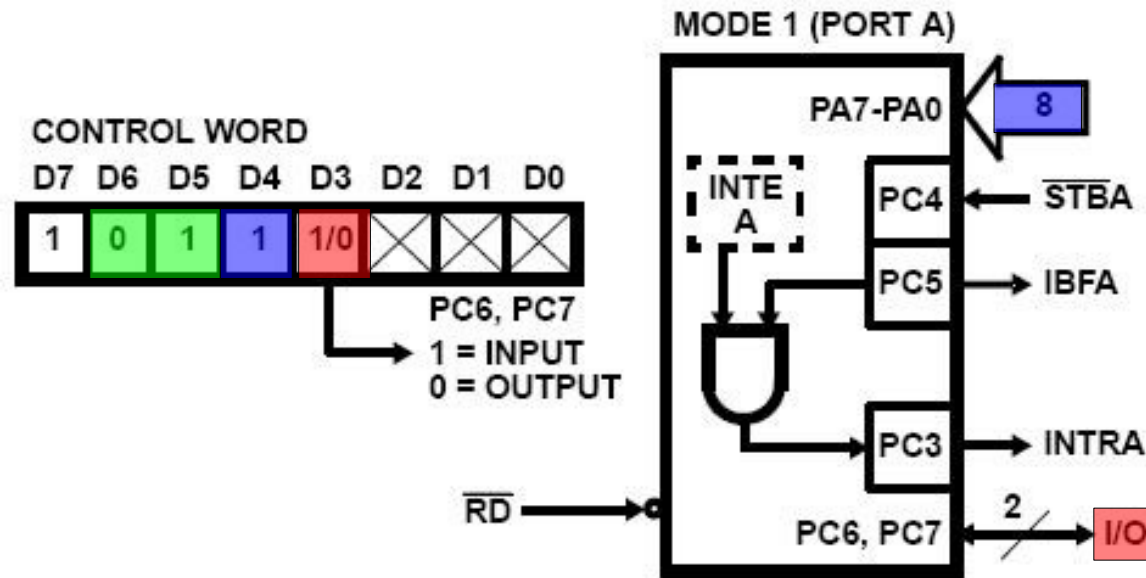
7. Daten gelesen AUS

0. Daten verfügbar AUS

0. Daten gelesen AUS

Mode 1 Input

● Kanal A



● Kanal B

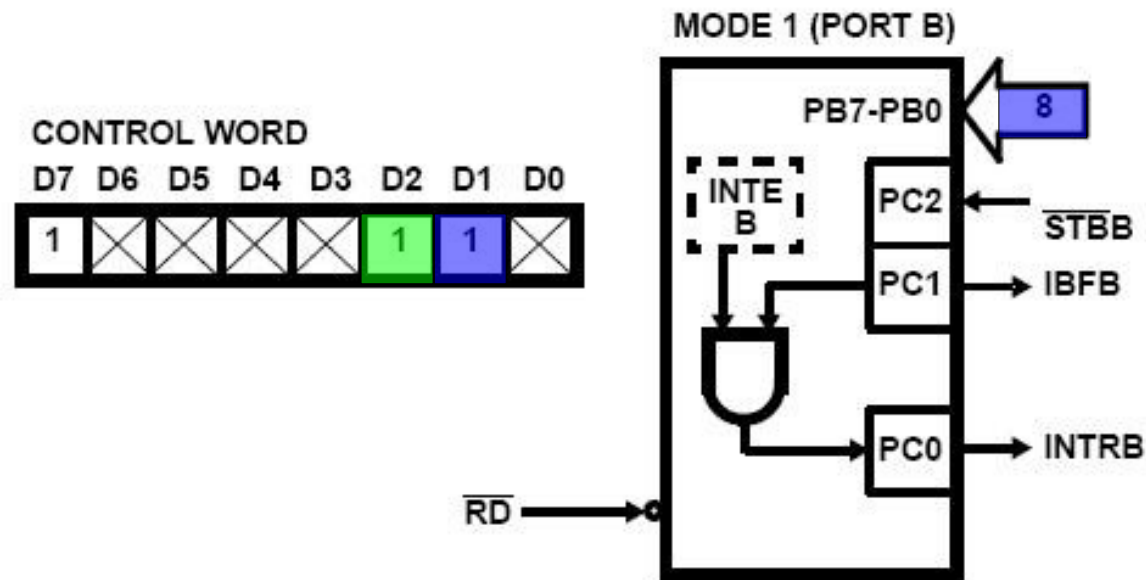
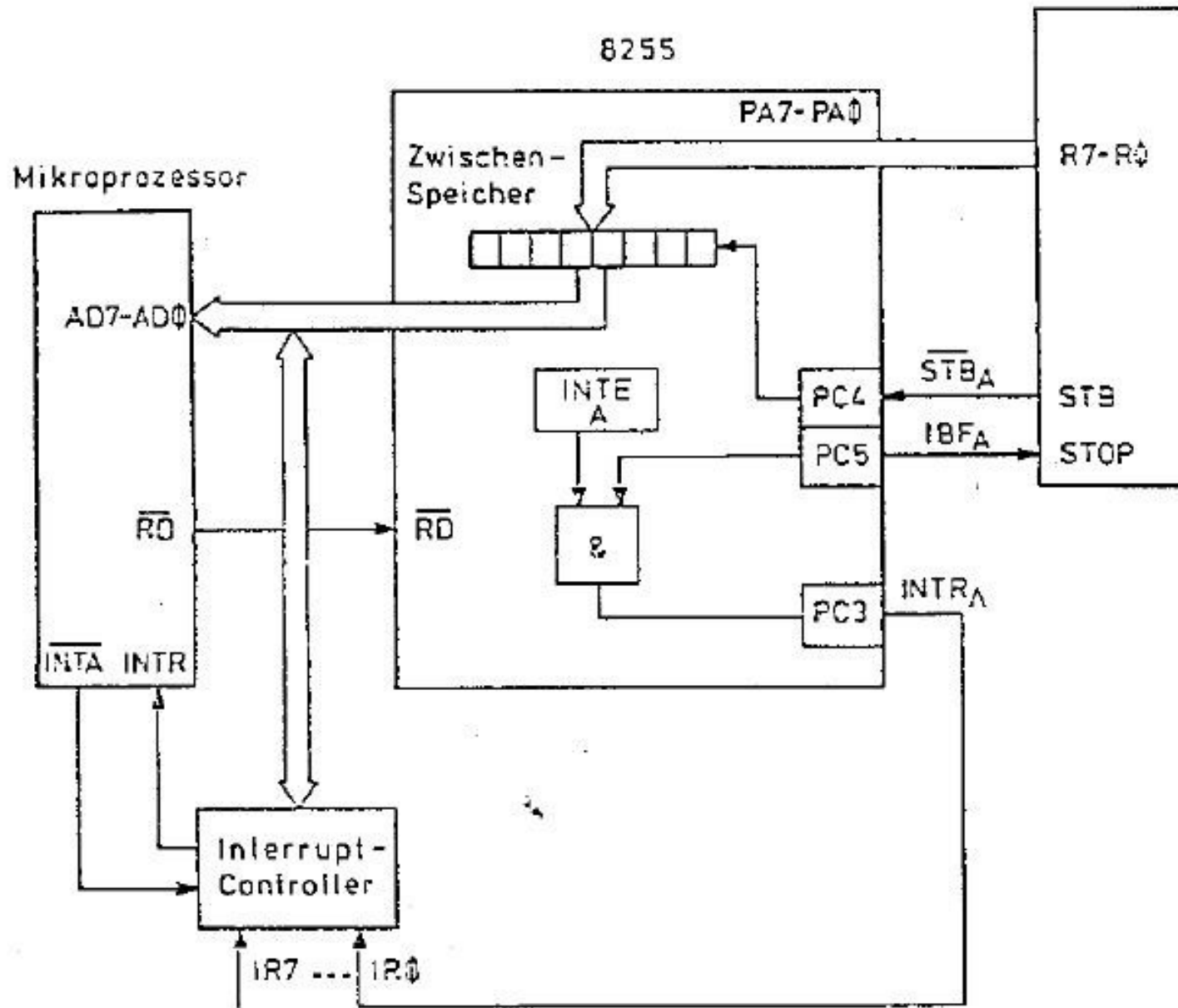


FIGURE 6. MODE 1 INPUT

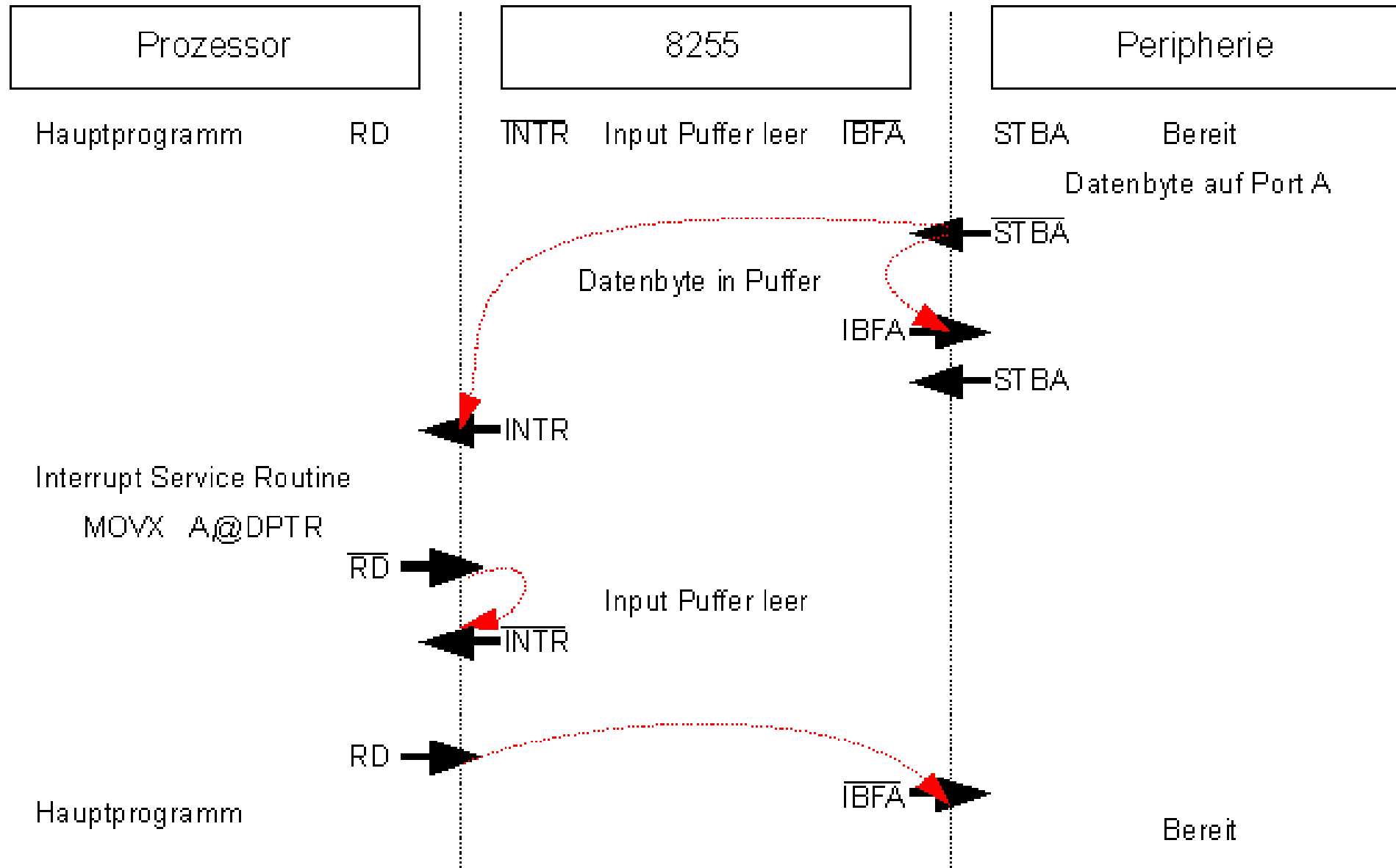
Mode 1 (für Kanal A und Kanal B)

- Zwei unidirektionale Schnittstellen für die 8-Bit Übertragung im Handshake-Verfahren.
 - Kanal A : Port A (Bits 7 ... 0) für Daten + Port C Bits 7 ... 3.
 - Inputoperation
 - Port C Bits 3 ... 5 für die Protokollsteuerung
 - Port C Bits 6 ... 7 für die Datenübertragung
 - Outputoperation
 - Port C Bits 3, 6 ... 7 für die Protokollsteuerung
 - Port C Bits 4 ... 5 für die Datenübertragung
 - Kanal B : Port B (Bits 7 ... 0) für Daten und Port C Bits 2 ... 0
 - Port C Bits 0 ... 2 für die Protokollsteuerung
- Schreibbefehle zu Port C für Kanäle in Mode 1 und Mode 2 sind wirkungslos.
 - Die betroffenen Port C Bits können nur mit Hilfe der Bit Set/Reset Funktion verändert werden.
- Lesebefehle von Port C für einen Kanal in Mode 1 und Mode 2 übertragen die Statusbits für den Kanal/die Kanäle.
- Kanal A in Mode 0 nur möglich, wenn auch Kanal B in Mode 0.

Mode 1 - Kanal A Input



Mode 1 Input Operation auf Kanal A



Timing : Mode 1 - Kanal A Input

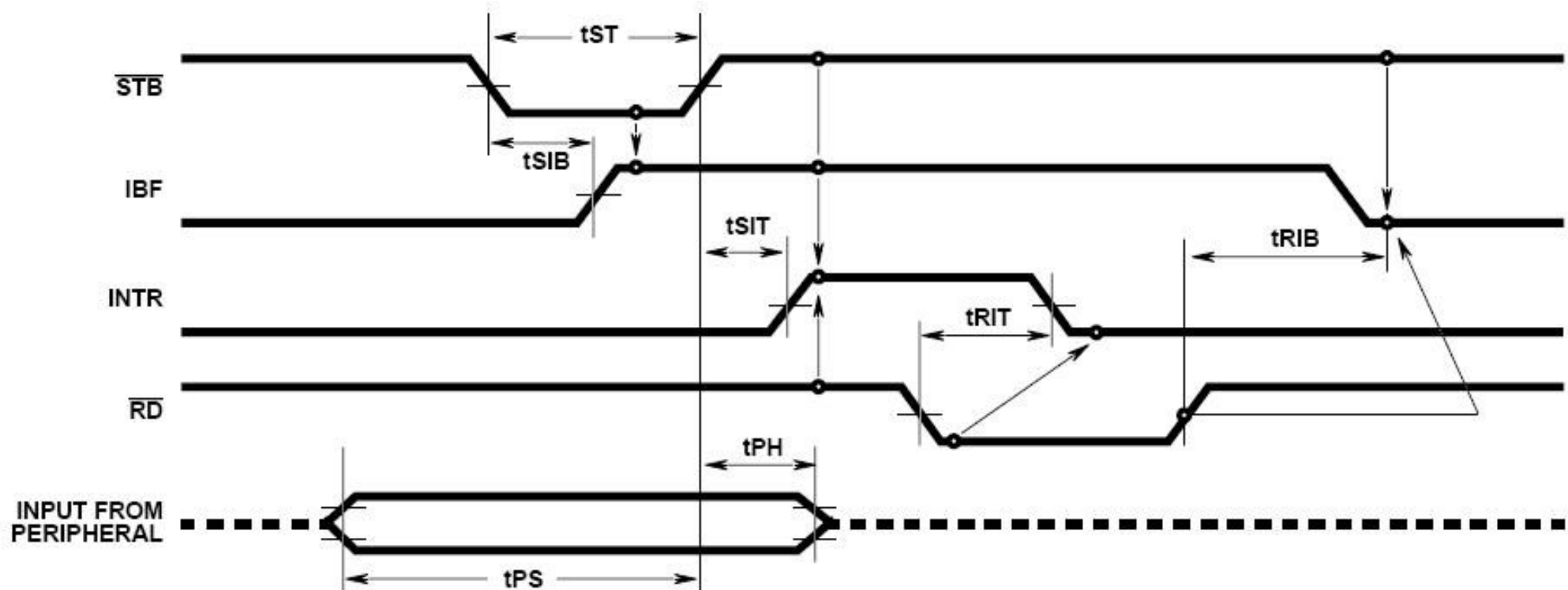


FIGURE 7. MODE 1 (STROBED INPUT)

Mode 1 Output

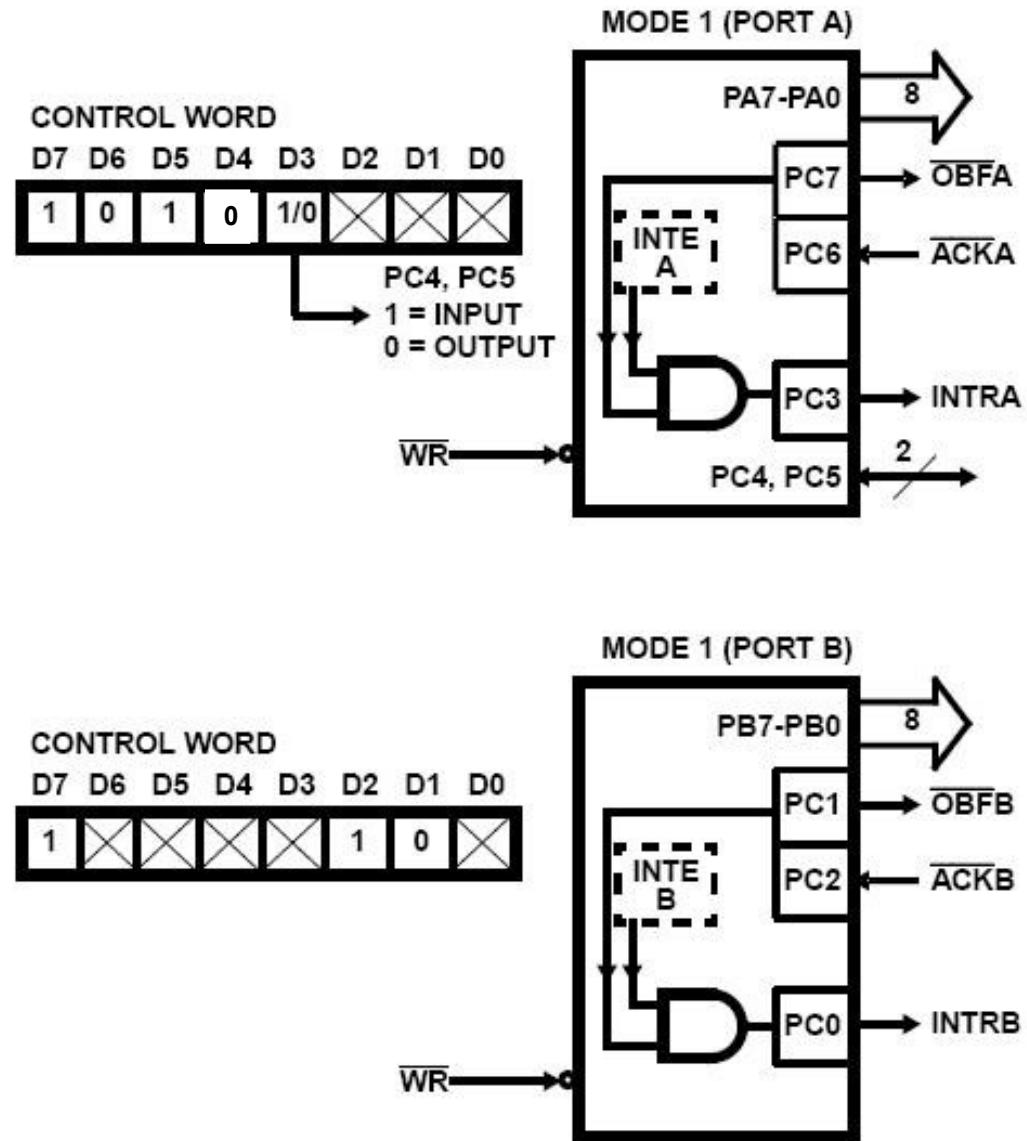


FIGURE 8. MODE 1 OUTPUT

Mode 2

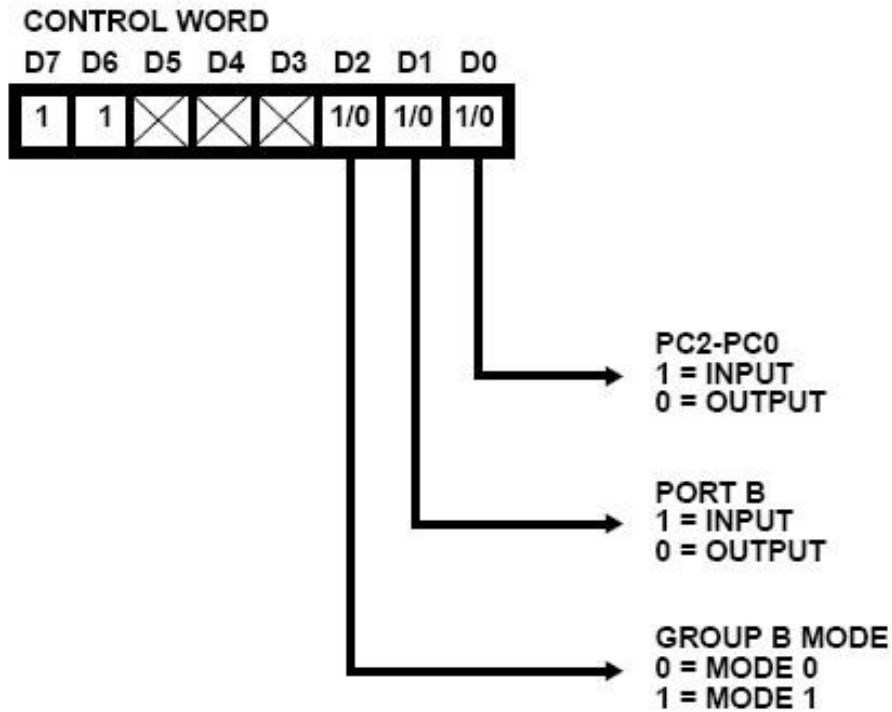


FIGURE 11. MODE CONTROL WORD

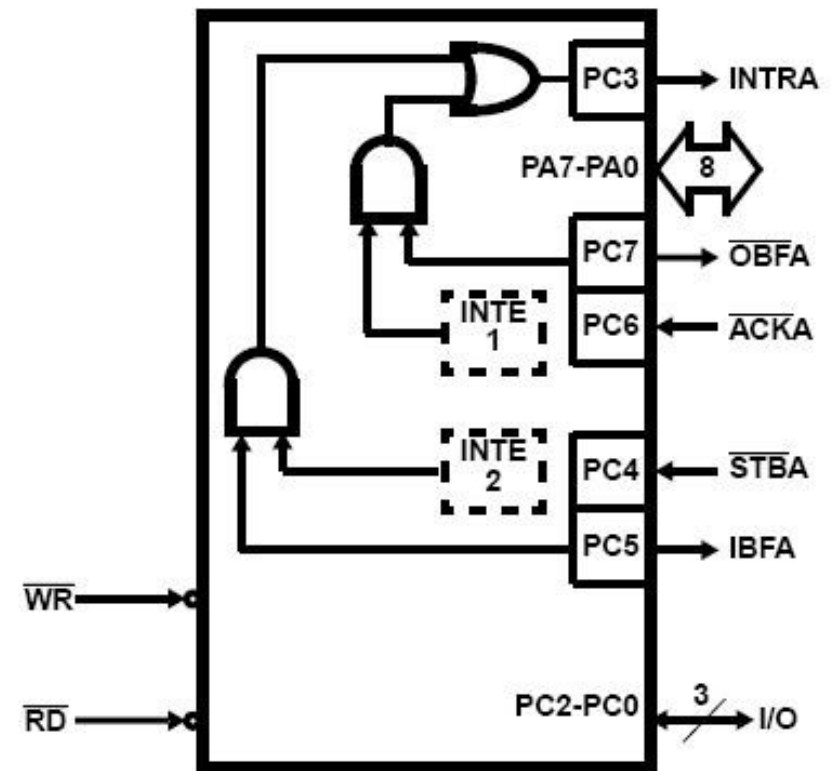


FIGURE 12. MODE 2

- Eine bidirektionale Schnittstelle für die Übertragung von 8 Bit Input- und 8 Bit Output-Daten.
 - Port A für Daten und Port C Bits 7 ... 3 für die Protokollsteuerung
 - Mode 2 nur auf Kanal A möglich.
- Kanal B kann zusätzlich in Mode 0 oder Mode 1 betrieben werden.

Mode 2 Statuswort

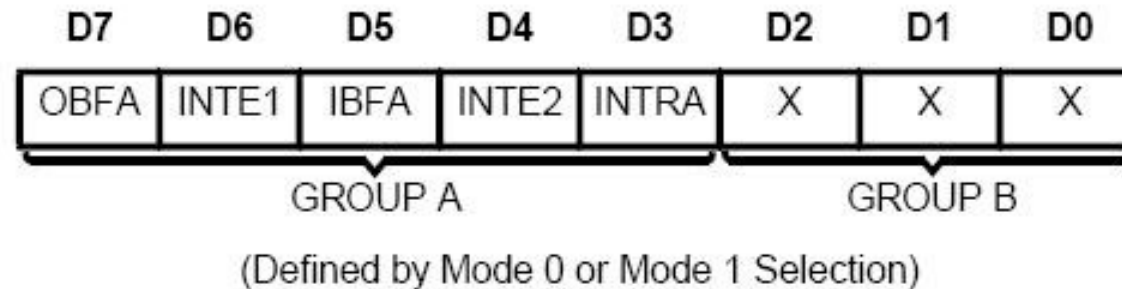


FIGURE 16. MODE 2 STATUS WORD FORMAT

Steuerung und Status in Mode 1 und Mode 2

- Steuerung
 - Steuerbits und Outputbits auf Port C können nur mit Hilfe der Port C Bit Set/Reset Funktion verändert werden.
 - Ein regulärer MOVX Schreibbefehl zu Port C ist wirkungslos.
 - Mit MOVX Befehlen für Port C werden nur die Port C Bits 0 ..2 übertragen, wenn Kanal B in Mode 0 arbeitet.
- Interrupt Request Signale zum Prozessor können innerhalb des 8255 Bausteins disabled werden.
 - Interrupt Enable Latches (INTE) können mit Hilfe der Port C Bit Set/Reset Funktion gesetzt oder zurück gesetzt werden.
 - Die Werte der Interrupt Enable Latches sind im Statuswort enthalten.
- Die Werte der Steuerleitungen von der Peripherie **zum** 8255 Baustein (\overline{ACK} und \overline{STB}) werden nicht im Status angezeigt.
 - Allerdings verursacht das Aktivieren einer dieser Leitungen sofort einen Interrupt, der wiederum im Statuswort angezeigt wird.
 - Somit ist auch ein Betrieb ohne Interrupts (Polling Mode) möglich.

Anwendungsbeispiel

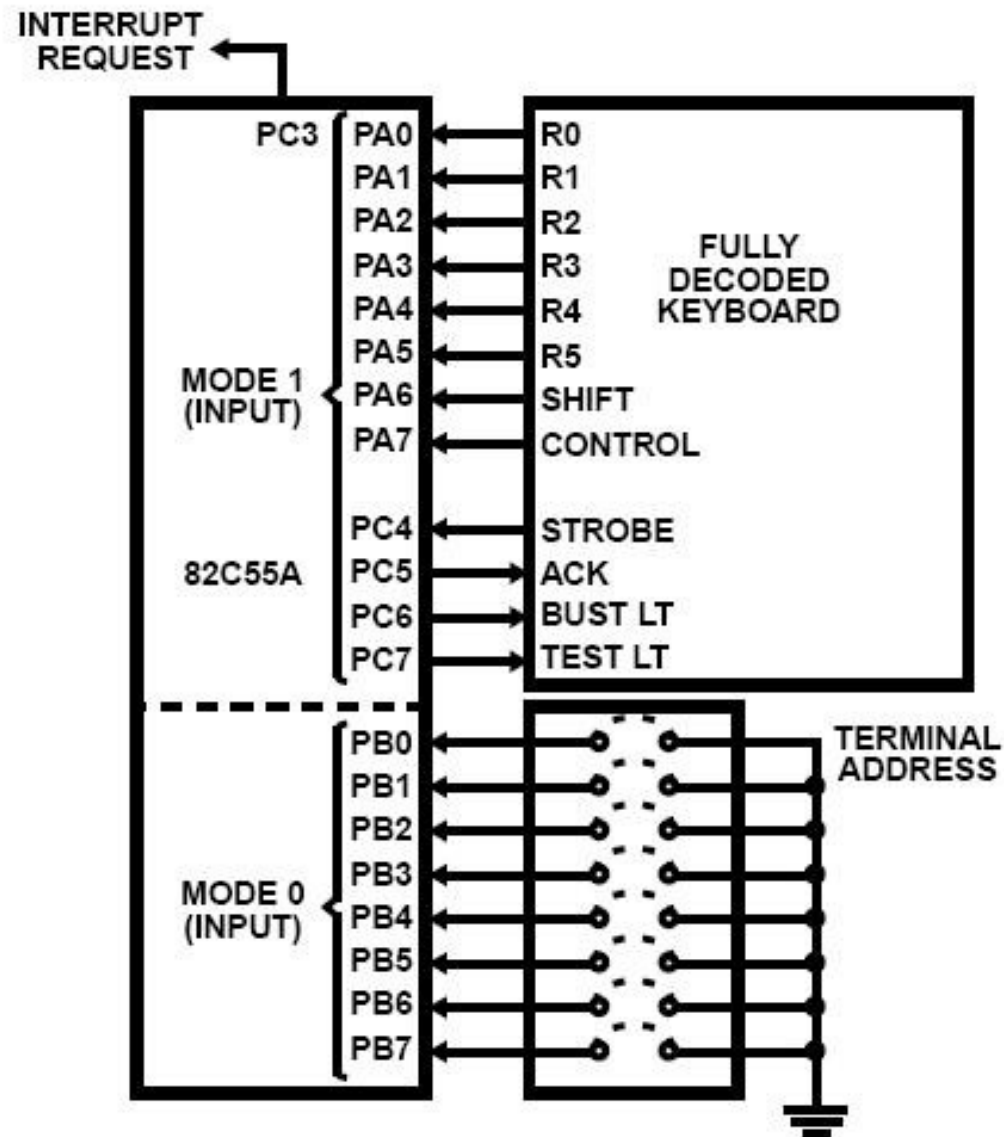


FIGURE 20. KEYBOARD AND TERMINAL ADDRESS INTERFACE