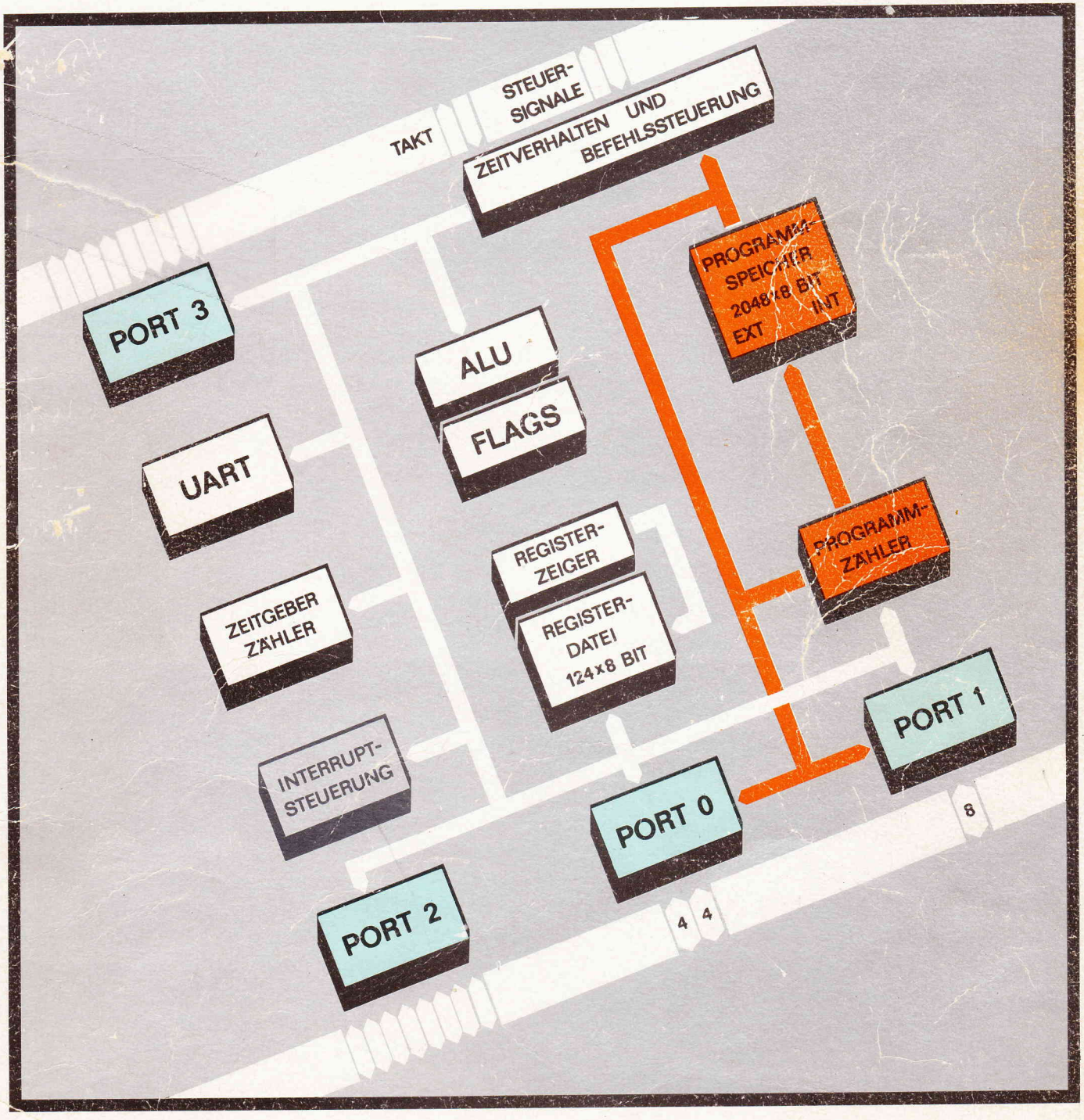
mikroelektronik

Einchip – Mikrorechner – Schaltkreise U 881 / U 882

RFT

Technische Beschreibung



Technische Beschreibung

Einchip - Mikrorechner - Schaltkreise

UB 8810 D
UB 8811 D
UB 8820 M
UB 8821 M

veb mikroelektronik ›karl marx‹ erfurt
stammbetrieb

DDR – 5010 Erfurt, Rudolfstraße 47 Telefon 580 Telex 061306

Die technischen Angaben dieses Handbuches tragen reinen Informationscharakter.
Verbindliche technische Liefer- und Reklamationsgrundlage sind ausschließlich die Typstandards.
Die vorliegende Dokumentation gibt keine Auskunft über Liefermöglichkeiten und beinhaltet keine Verbindlichkeiten zur Produktion.

Inhaltsübersicht

1. Einführung

Mit den Schaltkreisen UB 8810 D, UB 8811 D, UB 8820 M und UB 8821 M stehen dem Anwender leistungsfähige Einchip-Mikrorechner (EMR) zur Verfügung. Das grundlegend Neue dieser Einchip-Mikrorechner ist bereits aus Bild 1 ersichtlich.

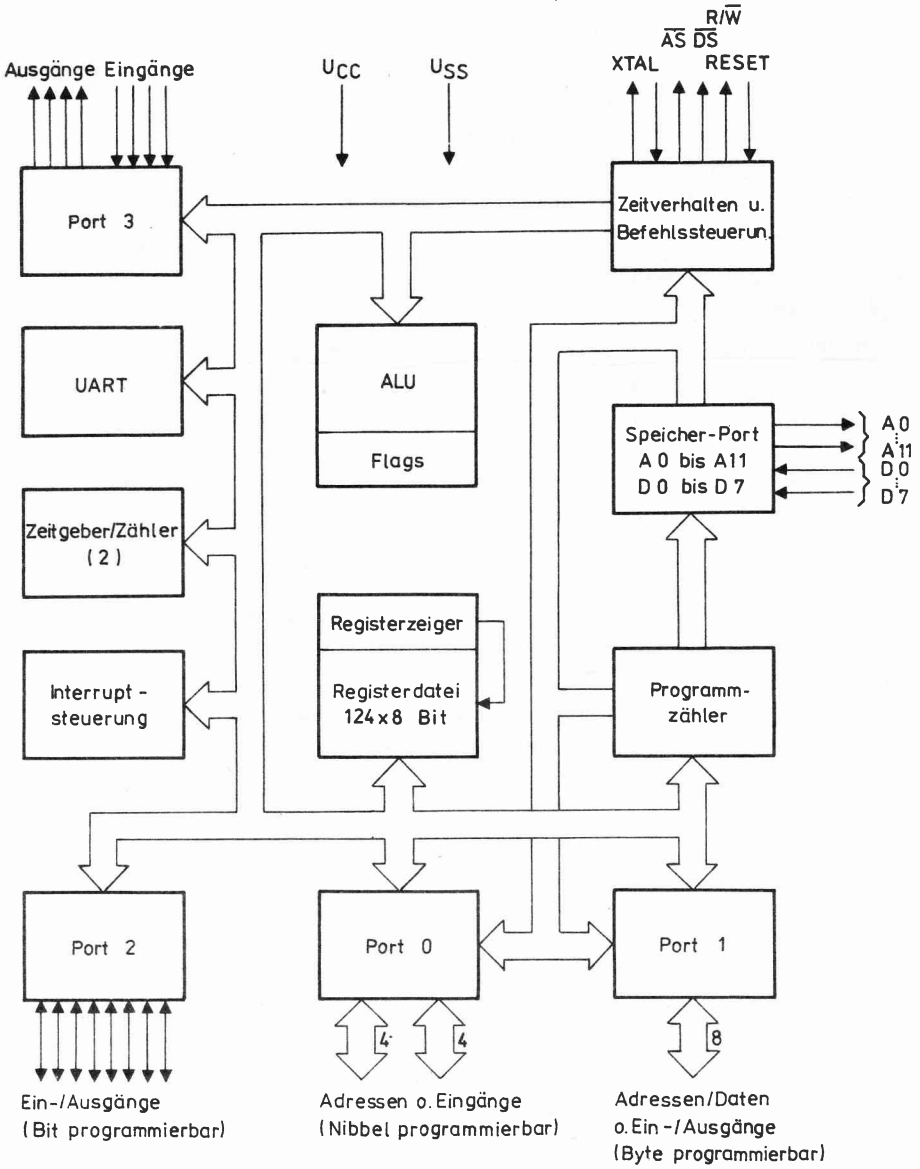


Bild 1: Blockschaltbild des EMR Beispiel UB 8820/UB 8821 M

Auf nur einem Chip, in einem einzigen Bauelement sind die wesentlichen Komponenten eines leistungsfähigen Mikrorechnersystems vereinigt.

- Programmspeicher (nur bei UB 8810 D/UB 8811 D)
- RAM
- Ports für parallele Ein-/Ausgabe
- Ports für serielle Ein-/Ausgabe
- Zähler/Zeitgeber
- Interruptsteuerung

Herausragende Eigenschaften der Einchip-Mikrorechner-Schaltkreise UB 8810,D, UB 8811 D, UB 8820 M, UB 8821 M sind:

- Verarbeitungsbreite	8 Bit
- Zahl der Basisbefehlstypen	43
- Speicherkapazität	2 KByte direkt adressierbar (bei UB 8820 M/ UB 8821 M, extern) bzw. als ROM (bei UB 8810 D/UB 8811 D, intern)

- RAM-Kapazität (intern) — 128 Byte (davon 124 Mehrzweckregister und 4 Ein-/Ausgaberegister dazu 16 Status- und Steuerregister)
- Ein-/Ausgabeleitungen — 32
- UART (vollduplex, durch internen Zeitgeber getaktet)
- 2 programmierbare 8-Bit-Zähler/Zeitgeber mit je einem programmierbaren 6-Bit-Vorteiler
- On-Chip-Oszillator (nur bei Anschlußvariante UB 88XO D/M)
- 6 priorisierte und vektorisierte Interruptquellen
- Möglichkeit der Adressierung externer Speicher bis 124 KByte
- Power-Down-Betriebsart (nur bei Anschlußvariante UB 88X1 D/M)
- TTL-Kompatibilität aller Anschlüsse

Die EMR-Schaltkreise UB 8810 D, UB 8811 D, UB 8820 M, UB 8821 M werden in n-Kanal-Silicon-Gate-Technologie gefertigt. Es werden jeweils zwei unterschiedliche Anschlußvarianten (Bondvarianten angeboten, die sich in der Belegung an Pin 2 bzw. 63 unterscheiden:

UB 8810 D, UB 8820 M — Es wird der On-Chip-Oszillator verwendet. Der direkte Anschluß eines externen Quarzes ist möglich.

UB 8811 D, UB 8821 M — Der Takt ist von einem externen Taktgenerator zu liefern. Es besteht die Möglichkeit zum Power-Down-Betrieb.

Anmerkung:

In der weiteren technischen Beschreibung wird generell die Abkürzung "EMR" verwendet, wenn sich nicht eine Unterscheidung zwischen der maskenprogrammierten (ROM-)Version UB 8810 D/ UB 8811 D und der Entwicklungsversion UB 8820 M/UB 8821 M notwendig macht. Desweiteren werden die Typbezeichnungen der EMR-Schaltkreise ständig mit dem Zusatzbuchstaben "B" (steht für 8 MHZ externe Taktfrequenz) verwendet, z. B. UB 8820 M, obwohl sich der Lieferumfang auch auf Typen mit geringerer externer Taktfrequenz erstreckt, z. B. UD 8811 D für 3,6 MHz. (siehe dazu auch Abschnitt 8. und 9.!)

2. Aufbau der EMR-Schaltkreise

2.1. Anschlußbeschreibung

2.1.1. Anschlußbelegung des UB 8810 D/UB 8811 D

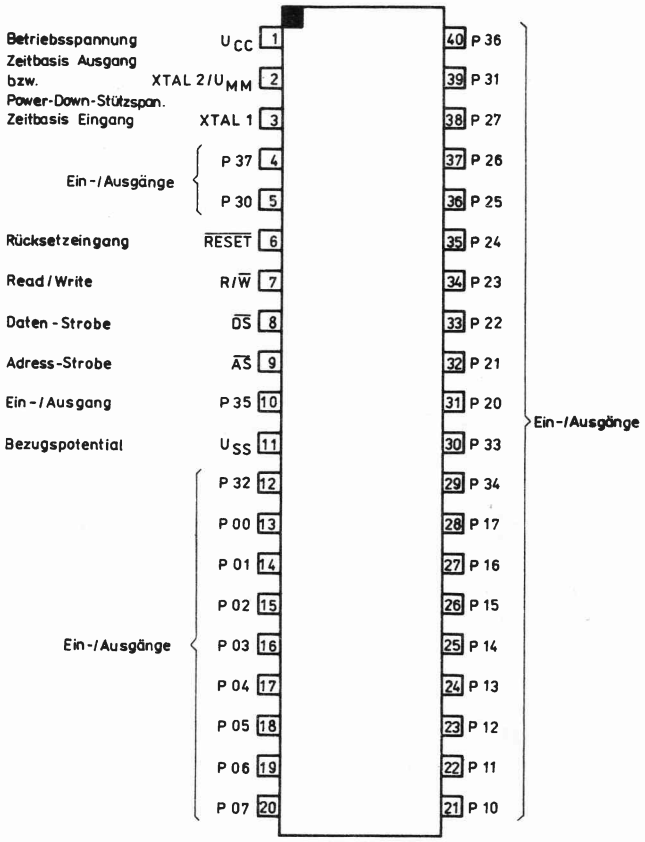


Bild 2: Anschlußbelegung des UB 8810 D/UB 8811 D

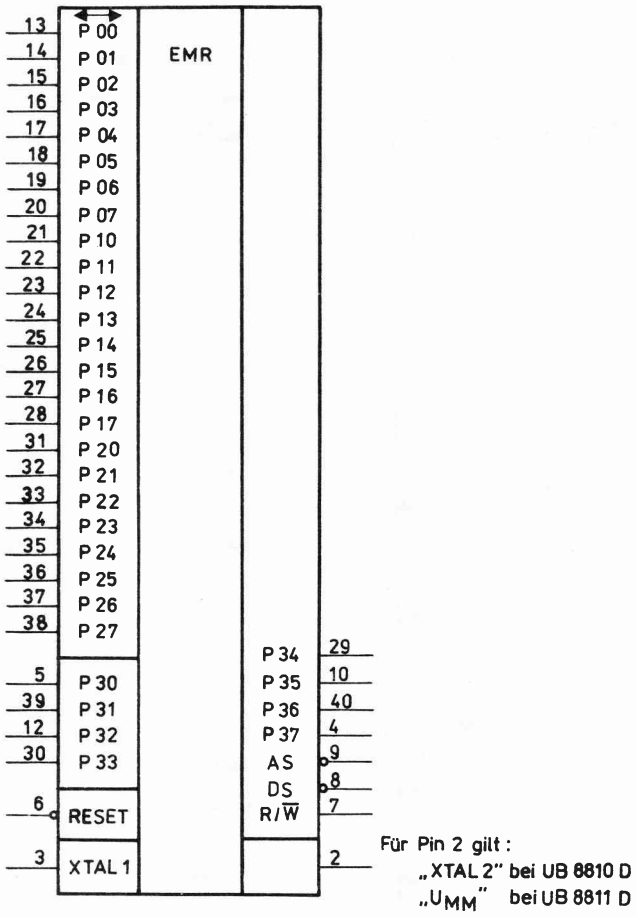


Bild 3: Schaltungskurzzeichen des UB 8810 D/UB 8811 D

2.1.2. Anschlußbelegung des UB 8820 M/UB 8821 M

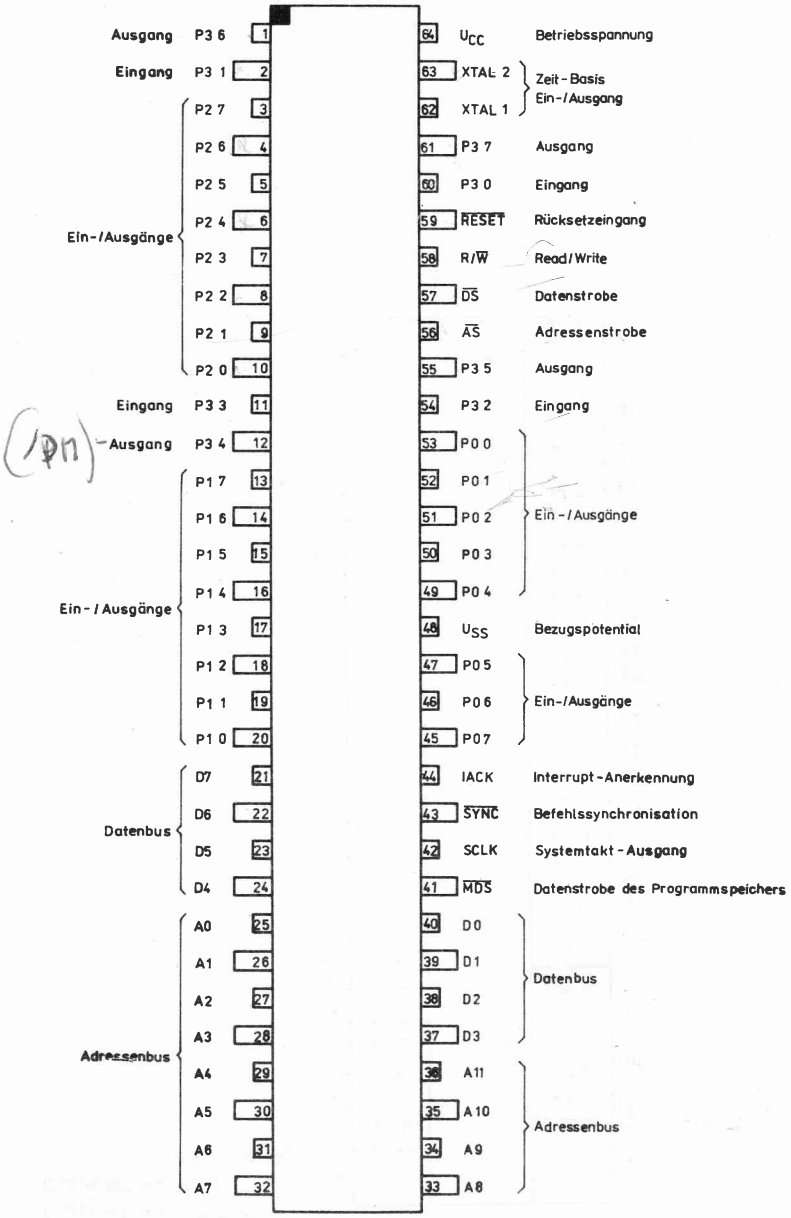


Bild 4: Anschlußbelegung des UB 8820 M/UB 8821 M

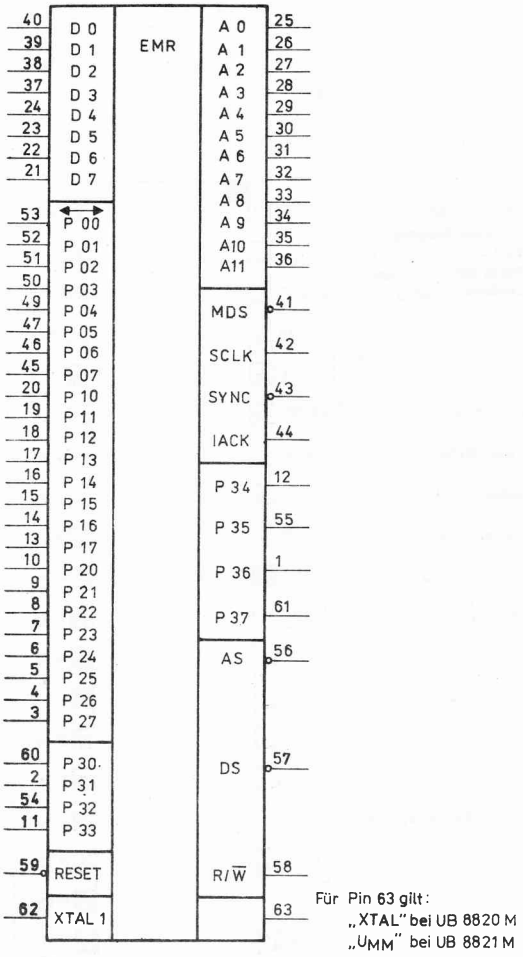


Bild 5: Schaltungskurzzeichen des UB 8820 M/UB 8821 M

2.1.3. Beschreibung der Anschlüsse

Nachstehende Anschlußbeschreibung gilt sowohl für die maskenprogrammierte (40 Pin-)Version als auch für die (64 Pin-)Entwicklungsversion des EMR. Auf einzelne Unterschiede wird an entsprechender Stelle im Text eingegangen.

P00 - P07
P10 - P17
P20 - P27
P30 - P37

= Ein-/Ausgabeleitungen (Ein-/Ausgänge, TTL-Kompatibel)
Diese 32 Leitungen sind in 4 Ein-/Ausgabeports zu je 8 Bit unterteilt, die durch Softwaresteuerung vielfältig konfiguriert werden können. Die einzelnen Leitungen eines Ports werden durch die zweite Ziffer gekennzeichnet, z. B. entspricht P 20 dem niedrigsten Bit von Port 2. Port 0 und Port 1 können zusätzlich zu ihren Ein-/Ausgabefunktionen durch Programmsteuerung als Interface für externe Speicher benutzt werden. Weiterhin kann Port 2 als "Open-Drain"-Ausgang konfiguriert werden.

$\overline{AS}$

= "Address Strobe" (Ausgang, Low-aktiv)
Dieses Signal erscheint, gepulst, sowohl beim Befehlsholezyklus aus dem internen und externen Programmspeicher, als auch beim Datentransfer vom und zum externen Datenspeicher. Die Adressen für alle externen Programm- und Datenübertragungen sind bei der (steigenden) Rückflanke von $\overline{AS}$ gültig. $\overline{AS}$ wird zu Beginn eines jeden Maschinenzyklus aktiv. Mit entsprechender Programmierung kann $\overline{AS}$ gemeinsam mit Port 0 und 1, $\overline{DS}$ und R/$\overline{W}$ in den hochohmigen Zustand versetzt werden.

$\overline{DS}$

= "Data Strobe" (Ausgang, Low-aktiv)
$\overline{DS}$ wird bei jedem externen Speichertransfer einmal aktiviert.
Schreibzyklus: EMR liefert gültige Daten am Port 1, während $\overline{DS}$ aktiv ist.
Lesezyklus: EMR empfängt gültige Daten an Port 1, während $\overline{DS}$ aktiv ist.
Mit entsprechender Programmierung kann $\overline{DS}$ gemeinsam mit Port 0 und 1, $\overline{AS}$ und R/$\overline{W}$ in den hochohmigen Zustand versetzt werden. Wenn der EMR nicht mit externem Speicher arbeitet, dient DS als Befehlssynchronsignal und wird während der Taktperiode, die dem Beginn des Opcode-Holens vorangeht, auf "Low gezogen.

R/$\overline{W}$

= "Read/Write" (Ausgang, Low-aktiv)
R/$\overline{W}$ ist "Low", wenn der EMR in den externen Speicher schreibt. Für alle anderen EMR-Zyklen bleibt R/$\overline{W}$ "High". Mit entsprechender Programmierung kann R/$\overline{W}$ gemeinsam mit Port 0 und 1, $\overline{AS}$ und $\overline{DS}$ in den hochohmigen Zustand versetzt werden.

XTAL 1
XTAL 2
Nur bei
UB 8810 D,
UB 8820 M

= "Crystal 1", "Crystal 2" (Zeitbasis, Ein- und Ausgang)
Diese Anschlüsse verbinden einen Schwingquarz (z. B. MQ 42/TGL 43 380) mit einem Serienresonanzwiderstand $\leq$ 100 Ohm oder einen externen Takt, gegenphasig an XTAL 1 und XTAL 2, mit dem On-Chip-Oszillator.

XTAL 1
U_{MM}
Nur bei
UB 8811 D,
UB 8821 M

= "Crystal 1", "Power-Down"-Stützspannung (Eingänge)
Beim "Power-Down"-Betrieb muß der EMR-Takt über XTAL 1 von einem externen Taktgenerator zugeführt werden.
Über den zweiten Eingang (sonst XTAL 2-Ausgang) wird die Stützspannung (U_{MM}) zugeführt, die bei U_{CC}-Ausfall die interne Registerdatei und Rücksetzlogik versorgt (siehe auch Abschnitt 7.1.).

$\overline{RESET}$

= "Reset" (Eingang, Low-aktiv)
$\overline{RESET}$ dient der EMR-Initialisierung und dem Schutz der Registerdatei während des Spannungszu- und abschaltens. Wenn $\overline{RESET}$ "High" wird, beginnt der EMR die Programmausführung, beginnend beim Programmspeicherplatz OOOOCH (siehe auch Abschnitt 7.2.). $\overline{RESET}$ wird auch benutzt, um den EMR in den Testbetrieb zu zwingen. Dies wird, durch Anheben der Spannung am $\overline{RESET}$-Eingang auf +7 V erreicht (siehe Abschnitt 7.4.).

A0 - A11
Nur bei
UB 8820 M,
UB 8821 M

= Programmspeicheradressen (Ausgänge),
A0 - A10 ermöglichen den Zugriff zu den ersten 2 KByte des Programmspeichers. Der Anschluß A11 bleibt vorerst reserviert für anderweitige Anwendungsfälle und ist nicht nutzbar.

D0 - D7
Nur bei
UB 8820 M,
UB 8821 M

= Programmdaten (Eingänge)
Über diese Anschlüsse erfolgt die Eingabe der durch A0 - A10 angewählten Daten aus dem Programmspeicher.

$\overline{MDS}$
Nur bei
UB 8820 M,
UB 8821 M

= "Memory Data Strobe" (Ausgang, Low-aktiv)
Während des Befehlsholezyklus ist $\overline{MDS}$ "Low", wenn auf die ersten 2 KByte des Programmspeichers zugegriffen wird, dagegen ist $\overline{MDS}$ während des Lesens eines Interruptvektors stets "High"!

$\overline{\text{SYNC}}$	= Befehlssynchronisation (Ausgang, Low-aktiv)
Nur bei UB 8820 M, UB 8821 M	Während der Taktperiode, die dem Beginn eines Befehlsholens vorausgeht, wird der Strobeausgang $\overline{\text{SYNC}}$ auf "Low" gesetzt.

SCLK	= Systemtakt (Ausgang)
Nur bei UB 8820 M, UB 8821 M	Über diesen Anschluß wird der interne Systemtakt gepuffert ausgegeben. Die interne Systemtaktfrequenz ist die halbe Quarzfrequenz!

IACK	= "Interrupt Acknowledge", Interrupt-Anerkennung (Ausgang, High-aktiv)
Nur bei UB 8820 M, UB 8821 M	Als Antwort auf einen Interrupt wird IACK während des Interruptmaschinenzyklus auf "High" geschaltet.

2.2. Beschreibung der Adreßräume und der Registerdatei

2.2.1. Programmspeicher

Der 16-Bit-Programmzähler adressiert 65536 Bytes des Programmspeicherraumes. Der Programmspeicher kann in 2 Bereichen angeordnet werden:

- intern: 2048 Bytes (2 KByte) - beim UB 8810 D/UB 8811 D als maskenprogrammierter ROM auf dem Chip, beim UB 8820 M/UB 8821 M außerhalb des Chips, über das Speicherport ansprechbar
- extern: 63488 Bytes (62 KByte) - über Port 0/1 bei entsprechender Konfigurierung ansprechbar

Die ersten 256 Bytes des externen Programmspeichers (Adressen 2048 bis 2048 + 255) können durch Konfigurieren des Tors 1 als zeitmultiplexes Adreß-/Datentor (AD0 - AD7) adressiert werden, das die Adreßbits A0-A7 und die Datenbits D0-D7 liefert. Tor 0 wird für zusätzliche 4 oder 8 Adreßbits (A8-A11 oder A8-A15) bei Anwendungen konfiguriert, die einen 4k- oder 64 K-Adreßraum des Programmspeichers erfordern.
Die ersten 12 Bytes des programmspeichers sind für die Interruptvektoren reserviert. Die Speicherzellen $00\text{-}0B_H$ (11_D) enthalten sechs 16-Bit-Vektoren, die mit den 6 möglichen Interrupts korrespondieren. Wenn ein Interrupt eintritt, wird die Programmsteuerung zu einer Serviceroutine übergehen, deren Adresse als Interruptvektor in den Zellen gespeichert ist, auf die sich der spezielle Interrupt bezieht. Ein "Rücksetzen" zwingt den Programmzähler auf den Stand $0C_H$ (12_D), d. h. die erste mögliche Adresse des Anwenderprogramms.

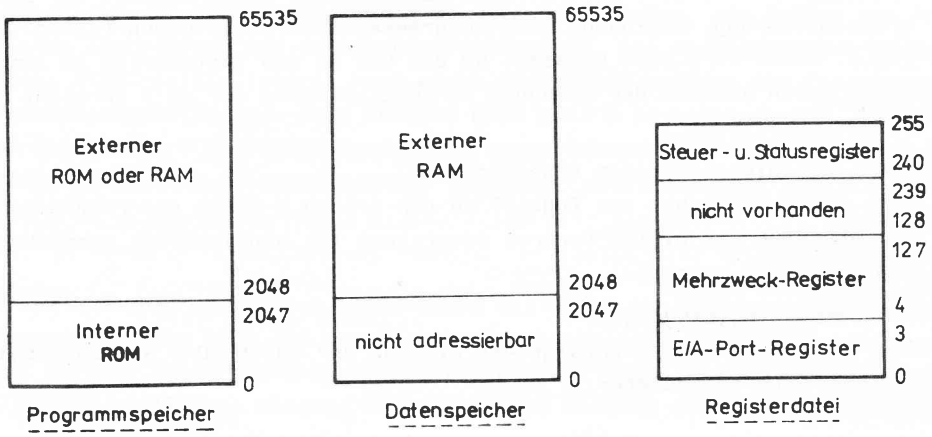


Bild 6 EMR-Adreßräume

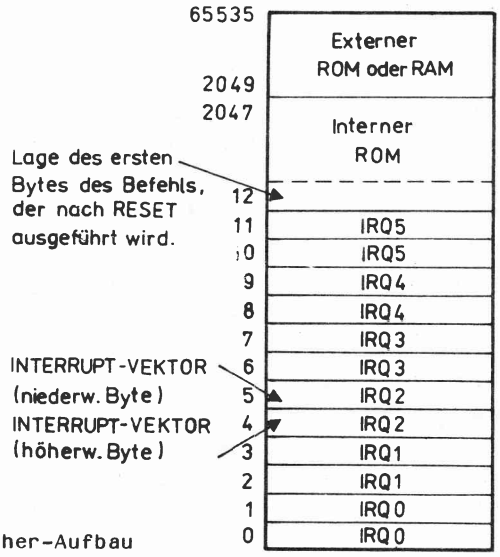


Bild 7: Programmspeicher-Aufbau

2.2.2. Datenspeicher

Ein EMR kann auf 62 KByte eines externen Datenspeichers zugreifen, beginnend mit der Speicheradresse 2048. Die Adressen werden dafür, wie beim externen Programmspeicher, über Port 0 und 1 bereitgestellt. Der externe Datenspeicher kann mit eingeschlossen oder getrennt vom externen Programmspeicher-Adreßraum angeordnet sein. Falls der Datenspeicher vom Programmspeicher getrennt wird, wird der Ausgang "Data Memory Select" ($\overline{DM}$) benutzt, um zwischen Datenspeicher und Programmspeicher auszuwählen.

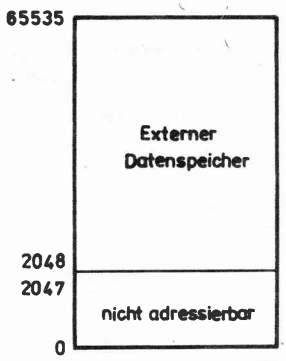


Bild 8: Datenspeicher-Aufbau

2.2.3. Externer Speicher

Bevor durch irgendeinen Befehl auf den externen Speicher zugegriffen werden kann, muß der Anwender die Ports 0 und 1 geeignet konfigurieren. Auf Grund des Befehlspipelining ist es unbedingt erforderlich, daß nach dem Festlegen der Betriebsart der Ports 0 und 1 für externe Speicheroperation, die nächsten 2 Bytes vom internen Programmspeicher geholt werden. Dafür können 2 Ein-Byte-Befehle, wie z. B. NOP's verwendet werden.
Die 2 externen Speicherräume, für Daten und Programm, können als ein einzelner Speicherraum von 62 KBytes oder als zwei getrennte Räume mit je 62 KBytes benutzt werden. Wenn der Speicherraum getrennt ist, werden Programmspeicher und Datenspeicher logisch durch den Ausgang "Datenspeicherauswahl" ($\overline{DM}$) getrennt. $\overline{DM}$ wird über Port 3, Leitung 4 (P34), durch entsprechende Programmierung des Port 3-Betriebsartenregisters geliefert. $\overline{DM}$ ist nur während der Ausführung der Befehle LDE, LDEI und der Befehle mit einer externen Kellerspeicherkonfiguration aktiv (CALL, PUSH, RETund IRET). Eine besondere Eigenschaft des EMR kann benutzt werden, um die Anzahl der Ein-/Ausgangsleitungen zu minimieren, die die Rolle der Adreßausgänge für mittelgroße Speicheranwendungen übernehmen. Diese Eigenschaft erlaubt dem Anwender, einen Speicher bis zu 10 KBytes mit nur 12 Adreßleitungen (plus den Steuerleitungen von $\overline{DM}$, $\overline{DS}$ und R/$\overline{W}$) zu adressieren. Normal würden 12 Adreßleitungen plus $\overline{DM}$ nur 4 KBytes in den beiden Programm- und Datenräumen adressieren

(in Wirklichkeit insgesamt 6 K, da die ersten 2K des Datenspeichers nicht adressierbar sind). Jedoch kann entweder $\overline{DS}$ oder R/$\overline{W}$ - im Effekt - eine 13. Adresse liefern. Dadurch werden, wenn der Anwendungsfall zwischen 4 K bis 6 K Programmspeicher (oder 2 K bis 4 K Datenspeicher) erfordert, nur Port 1 und das niedere Nibbel von Port 0 benötigt, die die Rolle der Adreßausgänge übernehmen. Wenn diese Eigenschaft nicht benutzt wird, muß das obere Nibbel des Ports 0 verwendet werden, um die zusätzlichen Adressen auszugeben.
Die folgende Tabelle stellt dar, wie der 4 K-bis 6 K - Adreßraum ohne ein 13. Adreßbit benutzt werden kann. Die Adreßleitungen A0-A11 sind ausreichend, um den internen 0K- bis 2K-Raum zu adressieren, wobei A11 immer "0" und $\overline{DS}$ und R/$\overline{W}$ inaktiv sind. A0-A11 sind erforderlich, um den 2 K- bis 4 K-Raum zu adressieren, wobei zu bemerken ist, daß $\overline{DS}$ und R/$\overline{W}$ nun aktiv sind. Für 4 K bis 6 K ist A11 wieder "0" (wie in 0 K- bis 2 K-Fall); jedoch, da $\overline{DS}$ und R/$\overline{W}$ noch aktiv sind, kann der 4 K- bis 6 K-Fall, vom 0 K- bis 2 K-Fall, wo diese Signale inaktiv sind, unterschieden werden.

Programmspeicheradresse (PC)	Datenspeicheradresse	Adresse auf Tor 0 und 1	A11	$\overline{DS}$ und R/$\overline{W}$
0-2047	-	0-2047	0	inaktiv
2048-4095	2048-4095	2048-4095	1	aktiv
4096-6143	4096-6143	0-2047	0	aktiv

Der 6 K- bis 8 K-Fall kann nicht vom 2 K- bis 4 K-Fall unterschieden werden, da in beiden Fällen $\overline{DS}$ und R/$\overline{W}$ aktiv sind und A11 gleich "1" ist. Deshalb muß das obere Nibbel vom Tor 0 benutzt werden, um Programm- und Datenspeicherräume größer als 6 K zu adressieren.

2.2.4. Registerdatei

Die 144-Byte-Registerdatei umfaßt 4 Ein-/Ausgabeportregister (R0-R3), 124 Mehrzweckregister (R4-R127) und 16 Steuer- und Statusregister (R240-R255). Den 144 Bytes der Registerdatei werden die in Bild 9 angegebenen Speicheradressen zugewiesen.

Anordnung der Register im RAM		Kurzzeichen
255	Stack Pointer (Bits 7 - 0)	SPL
254	Stack Pointer (Bits 15 - 8)	SPH
253	Register Pointer	RP
252	Programm Control Flags	FLAGS
251	Interrupt Mask Register	IMR
250	Interrupt Request Register	IRQ
249	Interrupt Priority Register	IRP
248	Ports 0 - 1 Mode	P01M
247	Port 3 Mode	P3M
246	Port 2 Mode	P2M
245	T0 Prescaler	PRE0
244	Timer/Counter 0	T0
243	T1 Prescaler	PRE1
242	Timer/Counter 1	T1
241	Timer Mode	TMR
240	Serial I/O	SIO
127	Mehrzweckregister	
4		
3	Port 3	P3
2	Port 2	P2
1	Port 1	P1
0	Port 0	P0

Bild 9: Organisation der Registerdatei

Die Ein-/Ausgabetore und Steuerregister sind in die Registerdatei eingeschlossen, um jeden beliebigen Befehl zu ermöglichen, Ein-/Ausgabe- oder Steuerinformation zu verarbeiten und dabei spezielle Ein-/Ausgabe- und Steuerbefehle zu vermeiden. Im allgemeinen können alle Mehrzweckregister als Akkumulatoren, Adreßzeiger oder Indexregister funktionieren. Bei der Befehlsausführung werden die Register gelesen, falls sie als Quellen definiert wurden und werden beschrieben, falls sie als Ziele definiert wurden.
Die Befehle greifen zu Registern direkt oder indirekt mit einem 8-Bit-Adreßfeld. Der EMR läßt auch 4-Bit-Registeradressierung unter Verwendung eines Registerzeigerverfahrens zu, das Bytes spart, die Programmausführungszeit verkürzt und die Programmumschaltung beschleunigt. (Die Programmumschaltung bezieht sich auf die Rettung und Rückspeicherung der Arbeitsregister, des Programmzählers, der Flage und der anderen zugehörigen Informationen, falls ein Interrupt eintrifft.)
In der 4-Bit-Adressierungsart wird die Registerdatei in 9 Arbeitsregistergruppen eingeteilt, die je 16 zusammenhängende Speicherzellen umfassen (Bild 9). Ein Register-Pointer (eines der Steuerregister) adressiert die Startspeicherzelle der jeweils aktiven Arbeitsregistergruppe. Jeder beliebige Befehl, der den Inhalt der Registerdatei verändern kann, kann verwendet werden, um den Register-Pointer zu variieren. Der EMR-Befehlssatz sieht auch einen speziellen Register-Pointer-Setzbefehl vor: SRP (Set Register Pointer).
Innerhalb der aktiven Arbeitsregistergruppe wird durch den 4-Bit-Register-Pointer, den der entsprechende Befehl selbst liefert, jeweils ein spezifisches Register festgelegt.

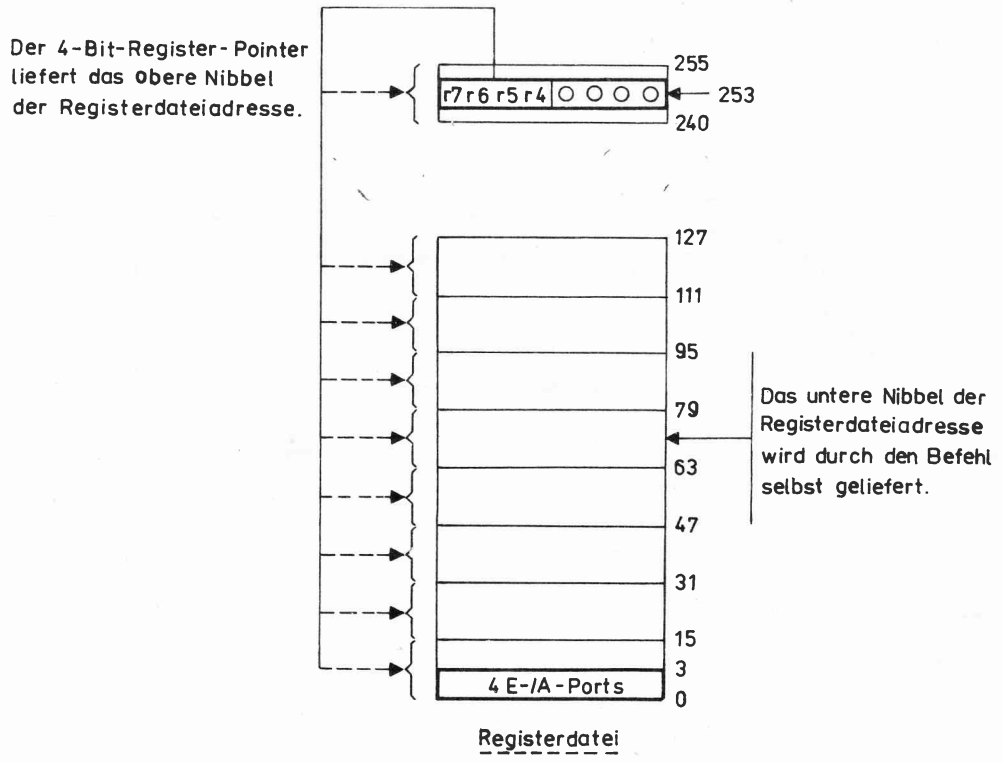


Bild 10: Wirkungsweise des Register-Pointers

2.2.5. Stack

Entweder die interne Registerdatei oder der externe Datenspeicher können als Stack (auch: "Kellerspeicher") verwendet werden. Die Auswahl wird durch die Programmierung eines Bits im Register R 248 vorgenommen. Ein 16-Bit-Stack-Pointer (R 254 und R 255) wird für den externen Stack verwendet, der irgendwo im Datenspeicher zwischen den Speicherzellen 2048 und 65535 liegen kann. Ein 8-Bit-Stack-Pointer (R 255) wird für die 124 Mehrzweckregister (R 4 bis R 127) benutzt.

Der Programmzähler während eines CALL-Befehls oder der Programmzähler und das Flagregister während eines Interruptzyklus, werden automatisch in den Stack gerettet. PUSH und POP-Befehle können jedes beliebige Register der Registerdatei retten und rückspeichern.
Eine Ausnahme bilden die "Nur-Schreibe"-Register. Die RET und IRET-Befehle speichern die geretteten Worte des Programmzählers bzw. des Flagregisters und Programmzählers zurück.

3. Arbeitsweise

In den folgenden Abschnitten soll die Arbeitsweise des EMR beschrieben werden, indem dessen Zeitverhalten beim "Befehls-Pipelining", beim Befehlszyklus, bei der Arbeit mit externem Speicher (bzw. Ein-, Ausgabe), beim Interruptzyklus und beim Rücksetzen erläutert wird.
Für die nachfolgenden Bilder gilt: Die Basiszeitperioden, die vom EMR benutzt werden, sind Maschinenzyklen (Mn), Zeitzustände (Tn) und Taktperioden. Alle Zeitbetrachtungen beziehen sich auf die Ausgangssignale $\overline{AS}$ und $\overline{DS}$. Der Takt wird nur zum Verständnis gezeigt und hat keine spezifischen Zeitrelationen zu anderen EMR-Signalen.

3.1. Befehls-Pipelining

Die relativ große Verarbeitungsgeschwindigkeit (Durchsatzrate) des EMR ist teilweise auf die Nutzung des "Befehls-Pipelinings" zurückzuführen. Dessen Prinzip basiert auf der Überlappung von Befehlsholezyklus und Ausführungszyklus, d. h. während der Ausführung eines Befehls wird der Opcode für den nächsten Befehl geholt (Bild 11).

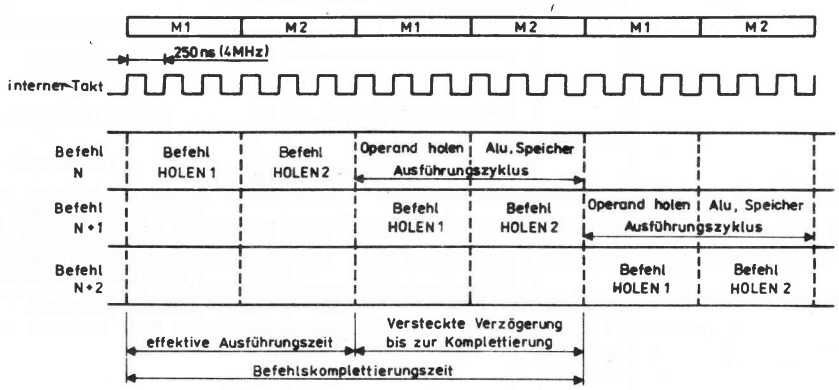


Bild 11: Befehls-Pipelining

Das Überlappen vom Holen des Befehls und der Ausführung bedingt, daß die "effektive Ausführungszeit" eines Befehls un den Betrag der Überlappung kürzer ist als die "Befehlskomplettierungszeit". Der Betrag der Überlappung wird "versteckte Verzögerung bis zur Komplettierung" genannt. Diese Verzögerung ist der Betrag der Zeit der durch den Befehl benötigt wird bis seine Ergebnisse gültig sind.
Wenn ein Programm läuft, ist die Befehlsüberlappungszeit vollständig in der gesamten Programmausführungszeit versteckt. Folglich kann dieser Zeitbetrag von der Befehlskomplettierungszeit subtrahiert werden, um die effektive Befehlsausführungszeit in einem Programm zu berechnen. Falls jedoch Ergebnisse von Einzelbefehlen zu testen sind, muß die versteckte Verzögerung zur Berechnung hinzu genommen werden.
Aufgrund des oben beschriebenen Pipelining-Effektes werden in Programmdurchsatzberechnungen nur die (effektiven) "Ausführungszyklen" verwendet. Da die (versteckten) "Pipeline-Zyklus" mit Ausnahme des letzten keinen Befehl beeinflußt, darf sie in Durchsatzberechnungen nicht verwendet werden.

3.2. Ablauf des Befehlszyklus

Bild 12 und 13 zeigen das Zeitverhalten des Befehlszyklus für das Holen der Befehle aus externen Speichern. Die Adressen $\overline{AS}$ und R/$\overline{W}$ werden bei Beginn eines jeden Maschinenzyklus (Mn) ausgegeben. Die Adressenausgabe über Tor 0 (wenn benutzt) bleibt über den Maschinenzyklus stabil, wohingegen die Adressenausgabe über Tor 1 nur während MnT1 gültig bleibt. Die Adressen werden mit der steigenden Flanke von $\overline{AS}$ garantiert gültig, die benutzt werden sollte, um die über Tor 1 ausgegebenen Adressen abzuspeichern. Tor 1 wird am Ende von MnT1 in den Eingabebetrieb gesetzt. $\overline{DS}$ wird während MnT2 ausgegeben, um zu ermöglichen, daß die Daten auf den Bus von Tor 1 gegeben werden. Der EMR akzeptiert die Daten, während MnT3, wenn $\overline{DS}$ beendet wird. Eine Taktperiode vor Beginn eines Opcodehole-Maschinenzyklus (M1) wird ein Befehlssynchronisationsimpuls ($\overline{SYNC}$) ausgegeben. Beim U 881 D/U 883 D geschieht dies nur (über das DS-Pin), wenn der externe Speicher nicht benutzt wird.

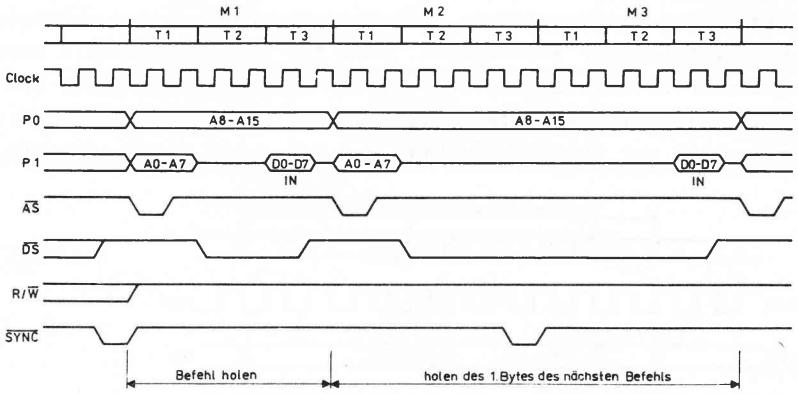


Bild 12: Zeitverhalten beim Befehlszyklus (Ein-Byte-Befehl)

Zu bemerken ist, daß alle Befehlsholezyklen dasselbe Maschinenzeitverhalten haben, unabhängig ob der Speicher intern ist oder nicht. Wenn der EMR auf externen Speicherzugriff programmiert ist, werden auch bei internen Speicherzugriff die Adressen über Port 0 und 1 ausgegeben: $\overline{DS}$ und R/$\overline{W}$ bleiben jedoch inaktiv. Wenn er nur für internen Speicher konfiguriert ist, werden die Tore 0 und 1 für Ein-/Ausgabe benutzt. $\overline{DS}$ gibt $\overline{SYNC}$ aus und R/$\overline{W}$ ist inaktiv.
Eine Ausnahme zu dem Zeitverhalten beim Befehlsholen bildet das Opcodeholen eines Befehls, der dem Holen eines Ein-Byte-Befehls folgt. Ein-Byte-Befehle erfordern 2 Maschinenzyklen zur Ausführung. Das Pipelining verursacht, das Holen des Opcodes einen Maschinenzyklus früher zu beginnen.

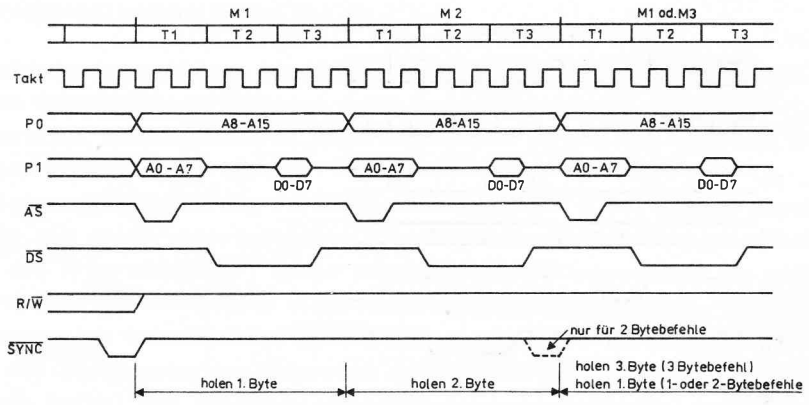


Bild 13: Zeitverhalten beim Befehlszyklus (2- und 3-Byte-Befehle)

3.3. Externe Speicher, Eingabe, Ausgabe

Falls der externe Speicher adressiert wird, werden die Tore 0 und 1 so konfiguriert, daß sie die erforderliche Anzahl von Adreßbits ausgeben. Tor 1 wird als gemultiplexter Adreß-/Datenbus für AD0-AD7 verwendet und Tor 0 gibt die Adreßbits A8-A15 aus. Die Zeitverhältnisse für die Adressierung externer Speicher und der Ein-/Ausgabe werden in Bild 14, 15, 16 und 17 dargestellt. Der Hauptunterschied zwischen diesen Darstellungen ist, daß Bild 16 und 17 einen hinzugefügten Zeitzyklus (Tx) enthält, der den Zeitverlauf beim externen Speicher erweitert, um auch den Einsatz langsamerer Speicher zu ermöglichen.
Die Adreßbits A0-A15 sind bei der steigenden Flanke von $\overline{AS}$ bei Speicherlese- und Speicherschreibzyklen gültig. Da Tor 0 nicht gemultiplext wird, stehen die Adreßbits A8-A15 - wenn benutzt - für den vollständigen Speicher-Lese-/Schreibzyklus bereit.
Während des Lesezyklus müssen die Eingangsdaten an Tor 1 mit der Rück-Flanke von $\overline{DS}$ gültig sein. Der Ausgang zur Auswahl des Datenspeichers ($\overline{DM}$) wird benutzt, um den externen Datenspeicher oder externen Programmspeicher auszuwählen. Wenn P34 dafür ausgewählt wurde, ist $\overline{DM}$ während der Ausführung gewisser Befehle aktiv. Während der Schreibzyklen haben die Adreßausgänge dasselbe Zeitverhalten wie bei Lesezyklen. Die Ausgangsdaten jedoch sind gültig, sobald während eines Schreibzyklus der Zustand $\overline{DS}$ aktiv und R/$\overline{W}$ aktiv (Low) eintritt.

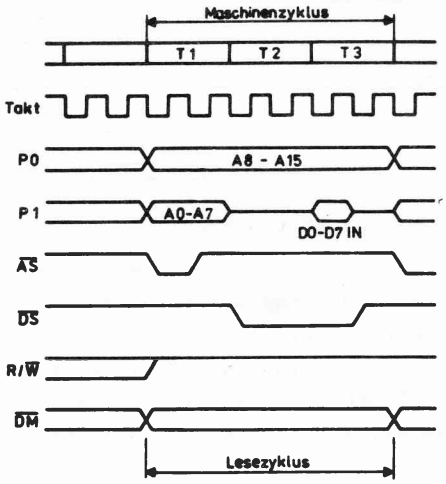


Bild 14: Holen externer Befehle, Ein-/Ausgabe oder Speicherlesezyklen

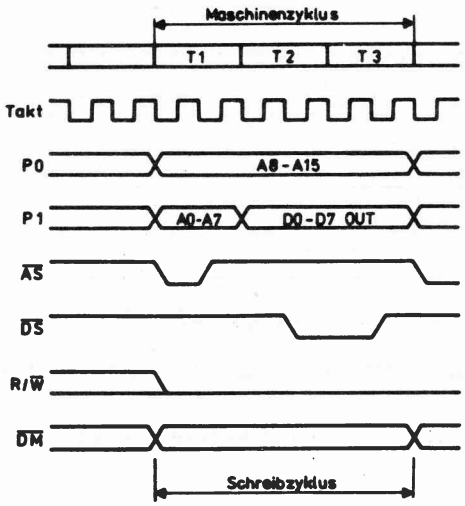


Bild 15: Externe Ein-/Ausgabe oder Speicherschreibzyklen

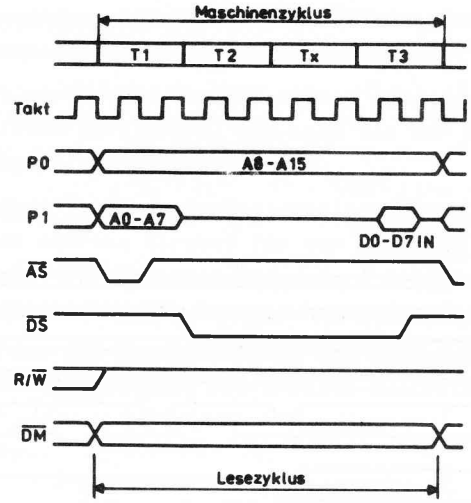


Bild 16: Erweitertes externes Befehlsholen, Ein-/Ausgabe oder Speicherlesezyklus

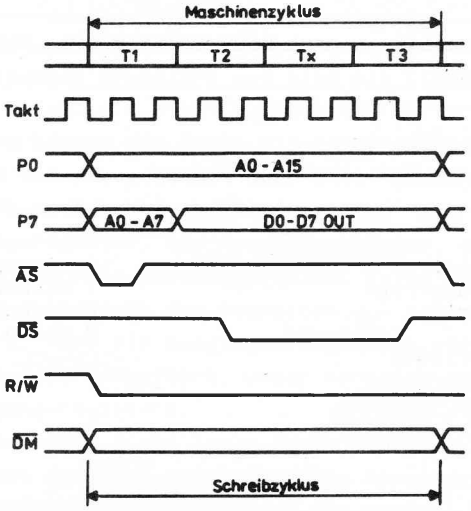
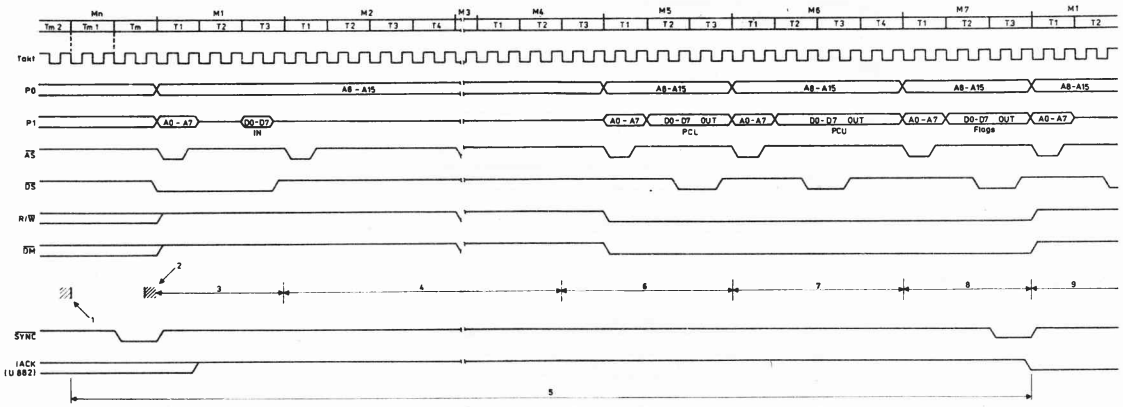


Bild 17: Erweiterter externer Ein-/Ausgabe- oder Speicherschreibzyklus

3.4. Interrrupt-Zeitverhalten

Interruptanforderungen werden vor jedem Befehlsholezyklus abgefragt (Bild 18). Zuerst werden die externen Interruptanforderungen 4 Taktperioden vor dem aktiven $\overline{AS}$-Impuls, der mit einem Befehlsholezyklus korrespondiert, abgefragt. Dann werden die internen Interruptanforderungen 1 Taktperiode vor $\overline{AS}$ abgefragt.
Wenn eine Interruptanforderung vorliegt, benötigt der U 881 sieben Maschinenzyklen (48 Taktperioden) um die Interruptprioritäten festzustellen, den richtigen Interruptvektor auszuwählen und den Programmzähler, sowie die Flags im Kellerspeicher zu sichern. Obwohl Bild 18 das Zeitverhalten bei Anwendung eines externen Stacks darstellt, wird bei internem Stack dasselbe Zeitverhalten benutzt. Die gesamte Interruptantwortzeit (inkl. externer Interruptabfragezeit) für einen externen Interrupt beträgt 52 Taktperioden bis zu dem Zeitpunkt, bei dem der erste Befehl der Interruptserviceroutine geholt wird.
Gleichzeitig mit IACK wird auch das Bit 7 des Interruptmaskenregisters rückgesetzt, wodurch weitere Interruptanforderungen verhindert werden. Beim Holen des 1. Befehls der Interruptserviceroutine wird im Interruptrequestregister das dem Interrupt entsprechende Bit gelöscht.



1 externe Interrupteingänge abgefragt
2 interne Interruptanforderungen abgefragt
3 Holen erstes Byte (Daten ignoriert)
4 interne Ausführung
5 Interruptantwortzeit = 52 Taktperioden
6 Sichern PCL im Stack
7 Sichern PCU im Stack
8 Sichern FLAGS im Stack
9 Holen des nächsten Befehls

Bild 18: Interrupt-Zeitverhalten

3.5. Rücksetz-Zeitverhalten

Die interne Logik wird während des Rücksetzens initialisiert, wenn der Reset-Eingang für mindestens 18 Taktperioden (Bild 19) auf "Low" gehalten wird.
Während der Zeit zu der $\overline{\text{RESET}}$ "Low" ist, wird $\overline{\text{AS}}$ mit der internen Taktrate ausgegeben. $\overline{\text{DS}}$ auf "Low" gezogen. R/$\overline{\text{W}}$ wird inaktiv und die Tore 0, 1 und 2 in den Eingabebetrieb gesetzt. Wenn $\overline{\text{AS}}$ und $\overline{\text{DS}}$ beide "Low" sind, ist das normalerweise eine sich gegenseitigeausschließende Bedingung; deshalb kann das Zusammentreffen von $\overline{\text{AS}}$ "Low" und $\overline{\text{DS}}$ "Low" als eine Rücksetzbedingung für andere Geräte benutzt werden.

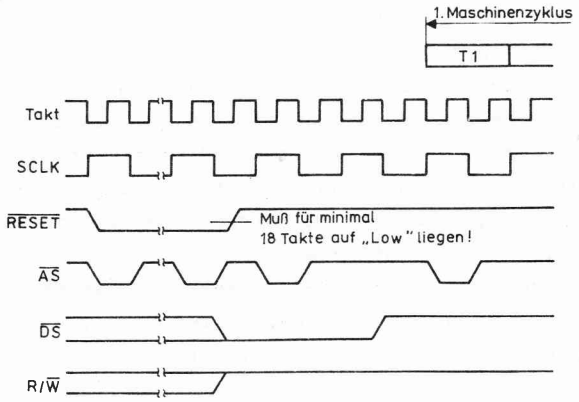


Bild 19: Zeitverhalten beim Rücksetz-Zyklus

3.6. Alternative Verwendung von Steuersignalen

Zusätzlich zu ihrer Anwendung bei Speicheroperationen können die Steuersignale $\overline{AS}$, $\overline{DS}$ und R/$\overline{W}$ bei folgenden Interfaceanwendungen genutzt werden:

$\overline{AS}$ - kann modifiziert werden, um das $\overline{RAS}$-Signal (Row Address Strobe) für den Anschluß dynamischer Speicher zu liefern. $\overline{RAS}$ kann von der Rückflanke von $\overline{DS}$ bis zur Rückflanke von $\overline{AS}$ abgeleitet werden.

$\overline{DS}$ - bietet verschiedene Verwendungsmöglichkeiten, z. B.:
- als $\overline{CAS}$ (Column Address Strobe) für den Anschluß dynamischer Speicher,
- als "Chip-Enable" für Speicher und andere Interface-Geräte,
- als Aktivierungseingang für 3-state-Bustreiber/-empfänger für Speicher und Interfaces.

R/$\overline{W}$ - kann benutzt werden als Schreibeingang zum Speicher-Interface und als frühzeitiger Statusausgang, um die Richtung von 3-state-Bustreibern/-empfängern schalten zu können.

3.7. Ein-/Ausgabe-Ports

In diesem Abschnitt werden Aufbau, Funktion und Arbeitsweise der Ein-/Ausgabeports beschrieben. Mit den Steuerregistern die zum konfigurieren dieser Ports benutzt werden, beschäftigt sich der Abschnitt 4.
Der EMR hat 32 Leitungen, die als Ein- und Ausgang verwendet werden können. Die Leitungen sind in 4 Ports zu je 8 Leitungen gruppiert und sind als Eingänge, Ausgänge oder Adreß-Datenleitungen konfiguriert.
Durch Softwaresteuerung können die Ports als Adreßausgänge für Timing als Statussignale und als serielle und parallele Ein-/Ausgabe mit oder ohne "handshake" vorgesehen werden.
Alle Tore haben mit TTL-Lasten kompatible "Pull-Ups' und "Pull-Downs". Jedes Bit der Ports 0, 1 und 2 hat ein Eingangsregister, ein Ausgangsregister, den dazugehörigen Puffer und die Steuerlogik. Bild 20 zeigt ein Port-Blockschaltbild. Falls ein Bit der Ports 0, 1 und 2 als Ausgang konfiguriert wird, veranlaßt das Schreiben des Bits, daß die Information im Ausgangsregister gespeichert wird. Wenn ein Ausgangsbit gelesen wird, wird die vorliegende Information des externen Anschlusses zurückgeführt. Unter normaler Ausgangsbelastung ist dies äquivalent mit dem Lesen des Ausgangsregisters.
Wenn jedoch ein Bit des Ports 2 als "Open-Drain"-Ausgang definiert wird, so muß die zurückgeführte Information nicht der Wert sein, den das Ausgangsregister enthält; vielmehr ist das der Wert, der auf den Eingangsanschluß durch das externe System aufgeprägt wird.

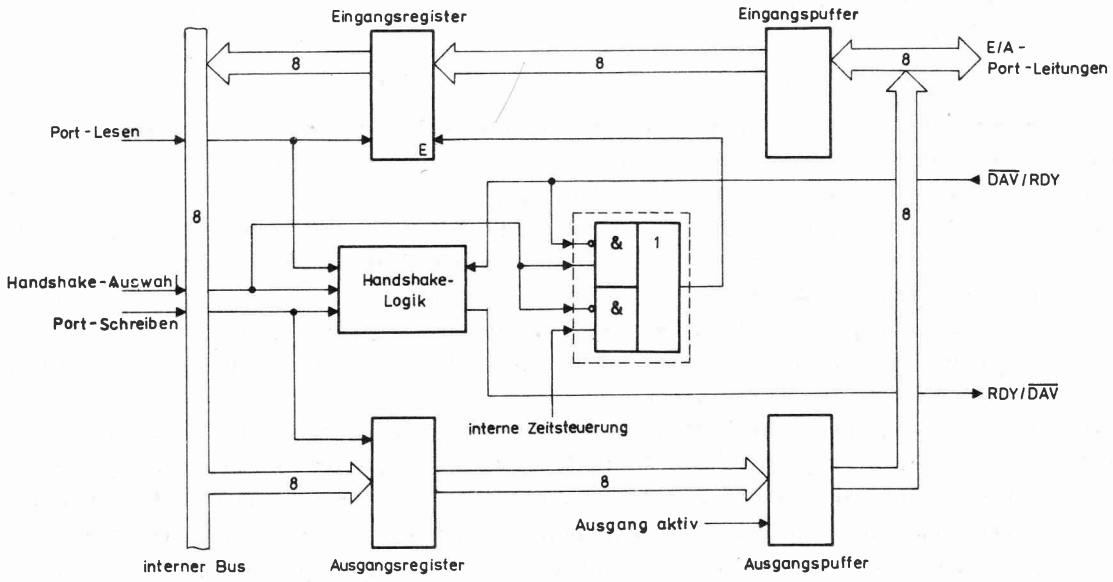


Bild 20: Blockschaltbild der Ports 0, 1 und 2

Falls ein Bit der Ports 0, 1 und 2 als Eingang definiert wird, veranlaßt das Lesen des Bits, daß die vorliegende Information am externen Anschluß gelesen wird. Die einzige Ausnahme sind Eingangsbits, die mit Handshake-Steuerung arbeiten. Beim Lesen eines Handshake-Eingangsbits wird die Information gelesen, die mit dem Strobe-Eingang in das Eingangsregister getaktet wurde. Eingangsbits können auch beschrieben werden, aber in diesem Falle werden die Daten im Ausgangsregister gespeichert und können nicht zurückgelesen werden. Wenn jedoch die Eingangsbits als Ausgangsbits rekonfiguriert werden, werden die in den Ausgangsregistern gespeicherten Daten an die Ausgangsanschlüsse geliefert. Dieser Vorgang erlaubt dem Anwender, die Ausgänge zu initialisieren, bevor sie ihre Ausgangslasten treiben.
Port 3 unterscheidet sich strukturell von den anderen Ports, da es nur ein einziges 4-Bit-Register besitzt, das mit seinen 4 Ausgangsbits verbunden ist. Falls in Port 3 geschrieben wird, werden die Daten im Ausgangsregister gespeichert. Falls Port 3 gelesen wird, werden die zurückgeführten Daten aus den Daten an den Eingangsanschlüssen und den Daten, die im Ausgangsregister gespeichert sind (nicht aus den Daten an den Ausgangsanschlüssen) zusammengesetzt.
Die Ausgänge von Port 3 können nicht beschrieben werden, wenn sie für solche Funktionen genutzt werden, wie serielle Ausgabe, Handshake-Steuerung oder Zeitgeberausgang.

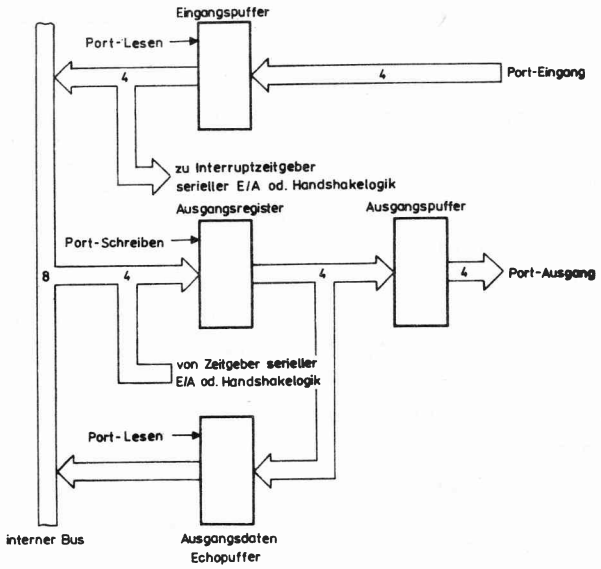


Bild 21: **Blockschaltbild von Port 3**

3.7.1. Port 1

Port 1 kann als Byteein-/ausgabetor mit oder ohne "Handshake", oder ein Adreß-/Datenport für das externe Speicherinterface programmiert werden. Die Konfiguration wird durch das Betriebsartenregister für Port 0 und 1 (P01M) R 248 festgelegt.
Falls es in der Betriebsart Byteeingabe oder Byteausgabe genutzt wird, wird das Port als allgemeines Register R1 behandelt. In das Port wird eingeschrieben, indem man R1 (der Registerdatei) als Zielregister des Befehles spezifiziert. Die Daten werden in dem Ausgangsregister des Ports gespeichert. Das Port wird gelesen, indem man R1 als ein Quellregister des Befehls spezifiziert. Die Daten werden in dem Ausgangsregister des Ports gespeichert. Das Port wird gelesen, indem man R1 als ein Quellregister des Befehls spezifiziert.
Falls es als ein Ein-/Ausgabeport benutzt wird, darf Port 1 für "Handshake"-Steuerung durch Programmierung des Betriebsartenregisters des Ports 3 (P3M) R 247 genutzt werden. In dieser Konfiguration werden die Anschlüsse des Ports 3 P33 und P34 als "Handshake"-Steuerleitungen $\overline{DAV1}$ für Eingangs-"Handshake" oder RDY1 und $\overline{DAV1}$ für Ausgangs-"Handshake" genutzt. Für Anwendungen mit externem Speicher, muß Port 1 auf die Betriebsart für zeitmultiplexe Adressen/Daten programmiert werden (AD0-AD7).

In dieser Konfiguration sind die unteren acht Bits der Adressen mit den Daten (DO-D7) zeitgeschachtelt. Verbunden mit Port 1 sind die Takt- und Steuersignale, Adreß-Strobe ($\overline{AS}$), Data Strobe ($\overline{DS}$) und Read/Write (R/$\overline{W}$). Bild 14, 15, 16 und 17 stellen die Zeitverhältnisse während der Speicherlese- und -schreiboperationen dar. In dieser Konfiguration können 2 zusätzliche Steuerleitungen - ein Interruptanforderungssignal als Eingang IRQ1 (P33) und das Signal $\overline{DM}$ (P34) für die Auswahl des externen Datenspeichers - benutzt werden.
Externe Speicherzellen größer als 2048 werden über Port 1 adressiert, wenn mehr als 256 externe Speicherzellen erforderlich sind, muß Port 0 zur Ausgabe der zusätzlichen Adressen genutzt werden.
Anmerkung:
Falls Port 1 als ein gemultiplextes Adreß-/Datentor konfiguriert ist, kann es nicht als ein Register behandelt werden. Zusätzlich zu den Ein-/Ausgabe- und Adreß-/Datenbetriebsarten, kann Port 1 in den hochohmigen Zustand (zusammen mit den Steuerleitungen $\overline{AS}$, $\overline{DS}$ und R/$\overline{W}$) versetzt werden, was dem EMR ermöglicht sich gemeinsame Quellen für Multiprozessor- und DMA-Anwendungen zu teilen. Diese Betriebsart funktioniert vollständig unter Softwaresteuerung und wird programmiert, indem man das P01M-Register (R248) benutzt. Datenübertragungen können durch logisches Zuweisen von P33 als ein Busbestätigungseingang BAK (IRQ1) und von P34 als ein Busanforderungsausgang BRQ gesteuert werden.

3.7.2. Port 0

Port 0 kann als ein "Nibbel"-Ein-/Ausgabeport oder als ein Adreßausgabeport zur Adressierung externer Speicher programmiert werden. Die Auswahl wird durch das Programmieren des Betriebsartenregisters für Port 0 und 1 (P01M) R248 getroffen.
Falls ein "Port 0-Nibbel" in der Ein-/Ausgabebetriebsart verwendet wird, wird es als korrespondierendes Nibbel des Registers R0 behandelt. Das Port wird eingeschrieben, indem man R0 als Zielregister eines Befehls spezifiziert und die Daten werden im Ausgangsregister des Ports gespeichert. Das Port wird gelesen, indem man R0 als Quellregister eines Befehls spezifiziert.
Anmerkung:
Ein Nibbel, das als Adreßausgang definiert ist, kann nicht als ein Register behandelt werden.
Falls es als ein Ein-/Ausgabeport verwendet wird, kann Port 0 unter "Handshake"-Steuerung durch Programmieren des Betriebsartenregisters (R3M) R247 des Ports 3 arbeiten. In dieser Konfiguration werden die Anschlüsse P32 und P35 als "Handshake"-Steuerleitungen $\overline{DAV0}$ und RDY0 für Eingangs-"Handshake" oder RDY0 und $\overline{DAV0}$ für Ausgangs-"Handshake" verwendet. Die "Handshake"-Signalzuweisung für die Leitungen P32 und P35 wird durch die Richtung (Eingang oder Ausgang), die dem oberen Nibbel des Ports 0 zugewiesen wurde, festgelegt.
Für Anwendungen mit externem Speicher kann Port 0 die Adreßbits A8-A11 (unteres Nibbel) oder A8-A15 (unteres und oberes Nibbel) liefern, je nach gefordertem Adreßraum.
Wenn der Adreßraum 12 Bits oder weniger erfordert, so kann das obere Nibbel von Port 0 unabhängig davon für Ein-/Ausgabe programmiert werden, während das untere Nibbel für die Adressierung benutzt wird. Falls die Port 0 - Nibbels als Adreßbits definiert werden, können sie gemeinsam mit Port 1 und den Steuersignalen $\overline{AS}$, $\overline{DS}$ und R/$\overline{W}$ durch Programmieren von (P1M) R248 in den hochohmigen Zustand versetzt werden.

3.7.3. Port 2

Die individuellen Bits von Port 2 können als Eingänge oder Ausgänge konfiguriert werden, indem man das Betriebsartenregister (P2M) R246 des Tores 2 programmiert.
Das Port wird als ein allgemeines Register R2 behandelt. So wie die Ports 0 und 1 wird in das Port, durch Spezifizieren von R2 als ein Zielregister eines Befehls, eingeschrieben und die Daten werden im Ausgangsregister gespeichert. Das Port wird gelesen, indem man R2 als Quellregister eines Befehls spezifiziert.
Port 2 kann unter "Handshake"-Steuerung durch Programmieren des Betriebsartenregisters (P3M)

R247 des Ports 3 arbeiten. In dieser Konfiguration werden die Anschlüsse des Ports 3 P31 und P36 als "Handshake"-Steuerleitungen verwendet: $\overline{DAV2}$ und RDY2 für Eingangs-"Handshake" oder RDY2 und $\overline{DAV2}$ für Ausgangs-"Handshake". Die "Handshake"-Signalzuweisung für die Leitungen P31 und P36 wird durch die Richtung (Eingang oder Ausgang), die dem Bit 7 des Ports 2 zugewiesen wurde, festgelegt.
Port 2 kann auch durch Programmierung des Betriebsartenregisters (P3M) des Ports 3 so konfiguriert werden, daß "Open-Drain"-Ausgänge entstehen.
Da alle externen Speicheradressen über die Ports 0 und 1 kommen, ist Port 2 immer für Ein-/Ausgabeoperationen benutzbar, sowohl für speicherintensive als auch für ein-/ausgabeintensive Konfigurationen.

3.7.4. Port 3

Die Leitungen von Port 3 können als Ein-/Ausgänge oder Steuerleitungen durch das Programmieren des Betriebsartenregisters von Port 3 (P3M) R247 konfiguriert werden. In jedem Fall ist die Richtung der 8 Leitungen als 4 Eingänge (P30-P33) und als 4 Ausgänge (P34-P37) festgelegt.
Port 3 wird als allgemeines Register R3 behandelt. Falls das Port gelesen wird, werden die gegenwärtigen Daten an den 4 Eingangsanschlüssen (P30-P33) und die im Ausgangsregister gespeicherten Daten (P34-P37) gelesen.
Die 4 Bits des Ausgangsregisters können nur beschrieben werden, wenn sie als Datenausgänge benutzt werden.
Für serielle Ein-/Ausgabe werden die Leitungen P30 und P37 als serieller Eingang bzw. serieller Ausgang programmiert. Details hierzu sind in Abschn. 3.7.5. zu finden.
Die Steuerfunktionen vom Port 3 (Tabelle 1) werden durch Programmieren des P3M-Registers definiert. Diese Steuerfunktionen können folgende Signale liefern:
- "Handshake" für Port 0, 1 und 2 ($\overline{DAV}$ und RDY)
- 4 externe Interruptanforderungssignale (IRQ0-IRQ3),
- Zeitgeber-Ein- und Ausgangssignale (T_{in} und T_{out}) und
- Auswahlleitung für externe Datenspeicher ($\overline{DM}$)

Funktion	Leitung	Signal
	P31	$\overline{DAV2}$/RDY2
	P32	$\overline{DAV0}$/RDY0
	P33	$\overline{DAV1}$/RDY1
Handshake	P34	RDY1/$\overline{DAV1}$
	P35	RDY0/$\overline{DAV0}$
	P36	RDY2/$\overline{DAV2}$
	P30	IRQ3
Interrupt-	P31	IRQ2
anforderung	P32	IRQ0
	P33	IRQ1
Zähler/	P31	T_{in}
Zeitgeber	P36	T_{out}
Statusausgang	P34	$\overline{DM}$

Tabelle 1: Steuerfunktionen von Tor 3

Zu beachten ist, daß die 4 Eingangsleitungen ohne Rücksicht auf die gewählte Konfiguration immer als Interruptanforderungsleitungen verwendet werden können. Ein Interrupt wird jedoch nur erzeugt, wenn das Interruptmaskenregister (IMR) R251 entsprechend programmiert wurde.

3.7.5. Port-Handshake

Die Ports 0, 1 und 2 können Daten übertragen, indem man die "Handshake"-Signale Ready (RDY) und Data Available DAV, Daten gültig) verwendet. Die Ausnutzung dieser Eigenschaft ist wählbar. Die "Handshake"-Signale sind so verschachtelt, daß die Datenübertragung asynchron arbeiten kann. Ein Leitungspaar von Port 3 (1 "Handshake"-Ausgang und ein"Handshake-Eingang") ist für jedes andere Port erforderlich, falls der "Handshake"-Betrieb benutzt wird. Das "Handshake"-Paar signalisiert die Funktion als "Bereit" (Ausgang) und "Daten gültig" (Eingang), falls das Port im Eingabebetrieb arbeitet oder als "Daten gültig" (Ausgang) und "Bereit" (Eingang), falls das Port im Ausgabebetrieb arbeitet.

Zur Beachtung: Der Anwender kann die "Handshake"-Ausgangsleitungen von Port 3 nicht (be)schreiben; die Ausgangs- und Eingangshandshakeleitungen können jedoch immer gelesen werden.

Empfehlung zur erstmaligen Aktivierung der "Handshake"-Folge:

1. - Port auf Ein-/Ausgabe setzen!
2. - Ausgangshandshakebit von Port 3 auf logisch "1" setzen!
3. - "Handshake"-Betriebsart für das Tor wählen!

Bild 22 und 23 beschreiben die detaillierte Arbeitsweise für die verschiedenen Phasen der "Handshake"-Folge. Beginnt einmal die Datenübertragung, so darf die Richtung der "Handshake"-Signale nicht geändert werden bis die "Handshake"-Folge komplettiert worden ist. Im Eingabebetrieb werden die Daten durch das erste "Daten gültig"-Signale in das Eingaberegister getaktet. Sie werden vor einem Überschreiben geschützt, falls zusätzliche Impulse auf dem DAV-Eingang einlaufen, bis die Daten gelesen werden. Das ist im Ausgabebetrieb nicht der Fall.

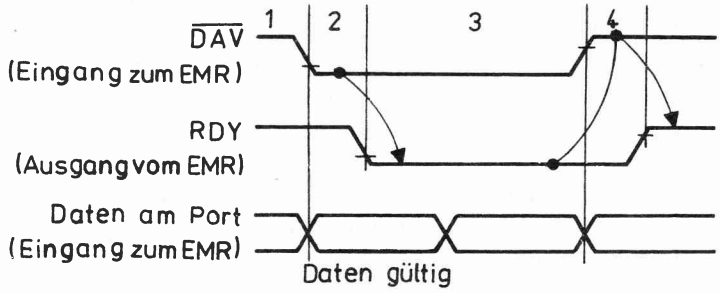


Bild 22: Eingabehandshakesignale

Erläuterungen zu Bild 22

Zustand 1: Ready-Ausgang von Port 3 ist "High", zeigt dadurch an, daß der EMR bereit ist, Daten anzunehmen.

Zustand 2: Das Ein-/Ausgabegerät liefert Daten an das Port und aktiviert den "Daten gültig"-Eingang ($\overline{DAV}$). Das erzeugt eine Interruptanforderung.

Zustand 3: Der EMR zwingt den Ready-Ausgang (RDY) auf "Low" und teilt dem Ein-/Ausgabegerät mit, daß die Daten eingetaktet worden sind. Das Ein-/Ausgabegerät kann dann $\overline{DAV}$ auf "High" ziehen. Der EMR muß auf die Interruptanforderung antworten und den Inhalt des Ports lesen, wodurch das "Handshake" komplettiert werden kann.

Zustand 4: RDY geht dann und nur dann auf "High", wenn das Port gelesen worden ist und $\overline{DAV}$ "High" ist.

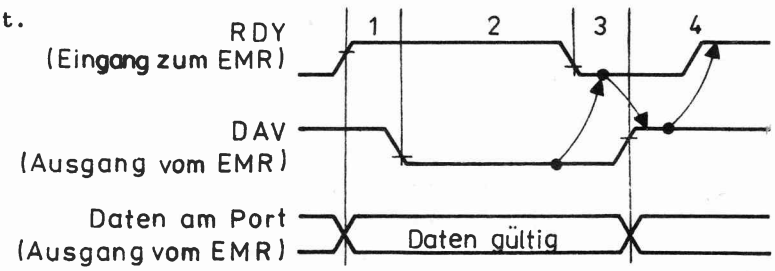


Bild 23: Ausgabehandshakesignale

Erläuterungen zu Bild 23

Zustand 1: Der EMR schreibt in das Portregister, um die Datenübertragung zu initialisieren. Wenn das Port eingeschrieben wird, wird dann und nur dann, wenn RDY "High" ist und die Ausgabedaten gültig sind, $\overline{DAV}$ "Low".

Zustand 2: Das Ein-/Ausgabegerät zwingt RDY auf "Low" nachdem es die Daten akzeptiert hat. RDY-"Low" bewirkt eine Interruptanforderung im EMR.

Zustand 3: Ein "Low" an RDY erlaubt dem EMR $\overline{DAV}$ auf "High" zu setzen.

Zustand 4: Nachdem $\overline{DAV}$ auf "High" geht, ist es dem Ein-/Ausgabegerät möglich auf RDY "High" zu gehen und damit zu signalisieren, daß es für die nächsten Daten bereit ist.

Im Ausgabebetrieb können die in ein Port geschriebenen Daten durch den EMR während der "Handshake"-Folge überschrieben werden. Damit ist es erforderlich, daß der Anwender einen Softwareschutz für die Daten vorsieht.
Bei Anwendungen, die einen Strobe-Ein- oder Ausgang für die Datenübertragung erfordern, kann der Anwender, statt der verschachtelten "Handshake"-Folge, die EMR-"Handshake"-Signalforderungen wie folgt erfüllen:

- Im Eingabestrobebetrieb kann der Anwender den Ready-Ausgang ignorieren und Daten laden, indem das $\overline{DAV}$-Signal benutzt wird, wobei eine Datenrate zulässig ist, die genügend Zeit für den U 881 läßt, die Daten anzunehmen, bevor die nächsten Datenzeichen geladen werden.
- Im Ausgabestrobebetrieb kann der Anwender den Ready-Eingang mit dem $\overline{DAV}$-Ausgang verknüpfen.

3.7.6. Serielle Ein-/Ausgabe

Die Leitungen P 30 und P 37 von Port 3 können als serielle Ein-/Ausgabeleitungen für voll-duplexen seriellen asynchronen Empfangs-/Sendebetrieb programmiert werden. Die Bitrate wird durch den Zähler/Zeitgeber 0 (T0) gesteuert und liefert eine maximale Datenrate von 62,5 KBit/s (f XTAL/128 bei max. 8 MHz). Die zu übertragenden Daten werden in Register R240 geladen und über P 37 hinausgeschoben (Bild 24). Die seriellen Daten werden über P30 empfangen, zu einem 8-Bit-Zeichen zusammengeführt und in den Empfangspuffer übertragen. Register 240 verhält sich in Wirklichkeit wie 2 Register: eingeschrieben wird in den Sender, gelesen wird aus dem Empfangspuffer.
Der T0 Zähler/Zeitgeber läuft mit dem 16fachen der Bitrate, um den ankommenden Datenstrom zu synchronisieren. Zur leichten Ableitung der allgemein benutzten asynchronen Datenkommunikationsbitraten kann ein 7,3728 MHz-Quarz für den EMR-Takteingang benutzt werden. Tabelle 2 listet die verschiedenen Bitraten und ihre erforderlichen T0-Initialwerte auf.
Im Sendebetrieb fügt der EMR automatisch ein Startbit und 2 Stopbits zu den gesendeten Daten hinzu. Der EMR liefert auch ungerade Parität, wenn das Steuerregister R247 (P3M) entsprechend programmiert wird. 8 Bits werden immer übertragen, ohne Rücksicht auf die Paritätswahl. Wird Parität aktiviert, ist das achte Bit das ungerade Paritätsbit. Zwischen den Zeichen wird der Ausgang P37 auf "High" gehalten, um die Markierungsbedingungen einzuhalten.

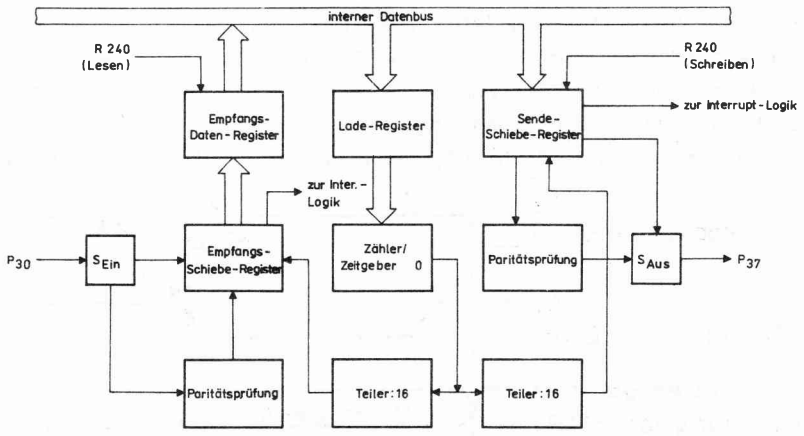


Bild 24: Blockschaltbild der seriellen Ein-/Ausgabe

Bitrate	TO-Anfangswert
19200	1
9600	2
4800	4
2400	8
1200	16
600	32
300	64
150	128
110	175 (Fehler 0,3 %)

Anmerkung: 7,3728; Quarz TO Vorteiler = 3

Tabelle 2: Ableitung von allgemein üblichen Bitraten

Bei Empfangsbetrieb muß das Datenformat ein Startbit, 8 Datenbits und mindestens 1 Stopbit haben (Bild 25). Wenn Parität gewählt ist, dann wird Bit 7 der empfangenen Daten (Paritätsbit) durch ein Paritätsfehlerflag ersetzt. Ein Fehler detzt das Flag auf logisch "1".

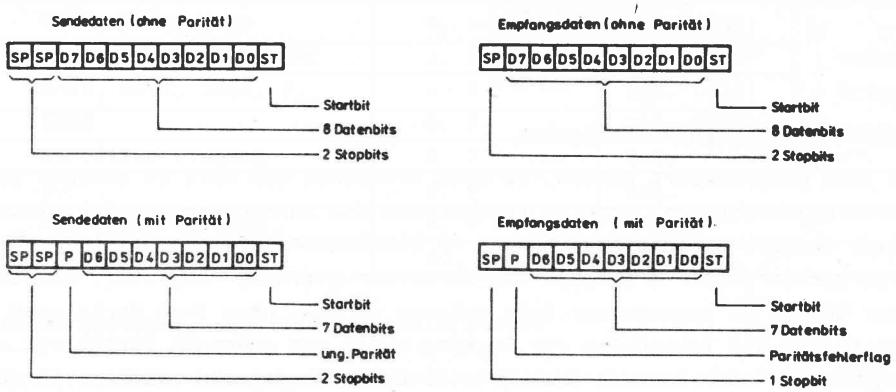


Bild 25: Serielle Datenformate

Eine Interruptanforderung (IRQ 3) wird jedesmal erzeugt, wenn ein Zeichen in den Empfangspuffer übertragen wurde. Obwohl der Empfänger doppelt gepuffert ist, ist er nicht vor einem Überschreiben geschützt. Ein gesendetes Zeichen erzeugt auch eine Interruptanforderung (IRQ 4), und wie der Empfangspuffer kann auch der Sendepuffer überschrieben werden.

3.8. Zähler/Zeitgeber

Der EMR enthält zwei 8 Bit-programmierbare Zähler/Zeitgeber (TO und T1), die jeweils durch einen eigenen 6 Bit-programmierbaren Vorteiler (Bild 26) getrieben werden. Der Vorteiler T1 kann durch interne und externe Taktquellen angesteuert werden; der Vorteiler TO wird nur durch den internen Takt angesteuert. Beide Zähler/Zeitgeber können unabhängig von der Prozessorbefehlsfolge arbeiten. Dadurch wird das Programm von zeitkritischen Operationen wie Ereigniszählung oder Zeitsummenberechnungen entlastet. Die Register R243 und R245 programmieren die 6-Bit-Vorteiler, um die Eingangsfrequenz der Taktquelle durch irgendeine Zahl von 1 bis 64 zu teilen.

Jeder Vorteiler steuert seinen Zähler (R244 für TO; R242 für T1) an, der den Wert (1 bis 256) dekrementiert, der in den Zähler geladen wurde. Falls der Zähler das Ende der Zählung erreicht, wird eine Interruptanforderung - IRQ4 bei TO oder IRQ5 bei T1 - erzeugt. Der Wert N sollte für eine Zählung von N geladen werden.

Die Zähler können gestartet, gestoppt, vom aktuellen Wert wieder gestartet oder vom Anfangswert aus gestartet werden, indem man das Betriebsartenregister der Zeitgeber (TMR) R241 programmiert.

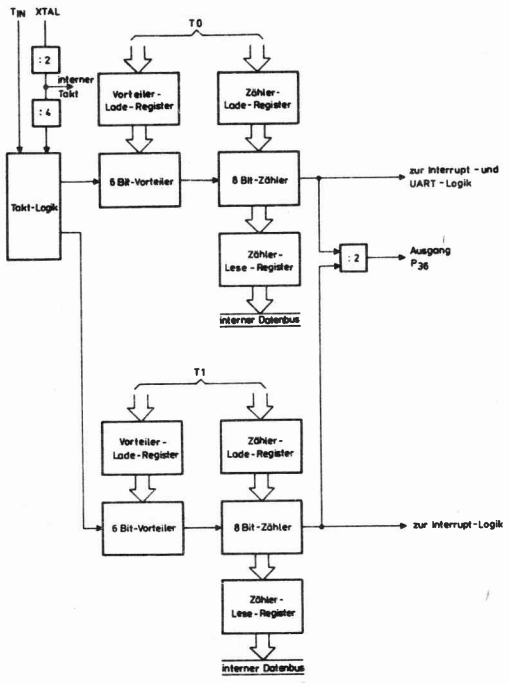


Bild 26: Blockschaltbild der Zähler/Zeitgeber

Die Zähler können auch programmiert werden, um beim Erreichen von Null zu stoppen (single-pass-mode = Einzeldurchlaufbetrieb) oder um automatisch den Anfangswert zurückzuladen und die Zählung fortzusetzen (modulo-n- continous mode = fortlaufenden Modulo-n-Betrieb). Das erfolgt durch Programmieren der Vorteiler 0 (PREO)- und Vorteiler 1 (PRE1)- Register. Die Zähler aber nicht die Vorteiler können zu irgendeiner Zeit gelesen werden, ohne ihre Werte oder die Zählbetriebsart zu zerstören. Die Taktquelle für T1 wird durch den Anwender festgelegt und kann der interne Mikroprozessortakt (max. 4 MHz) geteilt durch 4 oder ein externes Eingangssignal über Tor 3 (Leitung P 31) sein. Das Betriebsarteregister des Zeitgebers konfiguriert den externen Zeitgebereingang als einen externen Takt (max. 1 MHz), einen Triggereingang, der retriggerbar ist, oder als einen Toreingang für den internen Takt. Die Zähler/Zeitgeber können kaskadiert werden, indem man den Ausgang von T0 mit dem Eingang von T1 verbindet.
Die Leitung P36 von Port 3 dient auch als Zeitgeberausgang (T_{out}), durch welchen T0, T1 oder der interne Takt ausgegeben werden können. Der Zeitgeberausgang spricht am Ende der Zählung an. Wenn der Zeitgeber für den fortlaufenden Zählbetrieb programmiert wird, erzeugt P36 ein Ausgangstastverhältnis von 50 % T_{out} stellt sich neu ein, sooft neue Anfangswerte von den Laderegistern in T0 oder T1 entweder durch ein Softwareladekommando oder durch einen externen Triggereingang (nur T1) geladen werden.
Im Modulo-n-Zählbetrieb können neue Werte für beide Zähler in die Laderegister eingeschrieben werden, ohne daß die bereits laufende Abwärtszähloperation beeinflußt wird. Wenn das Ende der Zählung erreicht ist, wird der neue Anfangswert für die darauffolgenden Zähloperationen geladen.

3.9. Interrupts

Der EMR erlaubt 6 verschiedene Interrupts von 8 Quellen: die 4 Portleitungen P30-P33, serieller Eingang, serieller Ausgang und 2 Zähler/Zeitgeber. Diese Interrupts können maskiert und priorisiert werden, indem man das Interruptmaskenregister (IMR) R251 und das Interrupt-Prioritätsregister (IPR) R249 benutzt. Alle 6 Interrupts können global durch Rücksetzen des "Maskeninterruptenablebits" im Interruptmaskenregister (IMR) R251 unwirksam gemacht werden.
Alle EMR-Interrupts sind vektorisiert. Falls ein Interrupt eintrifft, geht die Steuerung zu einer Serviceroutine über, die angezeigt durch den spezifischen Programmspeicherplatz, für diesen Interrupt reserviert wurde. Diese Programmspeicherzelle und das nächste Byte enthalten die 16-Bit-Adresse der Interruptserviceroutine für diese einzelne Interruptanforderung.

Tabelle 3 zeigt die möglichen Interrupts, ihre Quellen, den Typ und die Vektorspeicherplätze. Weil TO den Takt für die SIO-Operationen liefert, schließen sich Interrupts von Zähler TO und von der seriellen Ausgabe gegenseitig aus. Beide verwenden die gleiche Interruptanforderungsleitung IRQ4. Entsprechend ist der Interrupt der seriellen Eingabe mit IRQ3 verbunden, weil die serielle Eingabe über P30 (IRQ3) erfolgt. Die 4 Eingabe-PIN's des Ports 3 sind in jedem Fall die Interruptanforderungseingänge IRQ0 - IRQ4. Ein High-Low-Übergang erzeugt an ihnen immer eine Interruptanforderung.
6 Bits im Interrupt-Maskenregister R251 können individuell die 6 Interruptanforderungen IRQ0 - IRQ5 wirksam oder unwirksam machen. Bit 7 macht global alle Interrupts unwirksam. Falls mehr als ein Interrupt ansteht, werden die Prioritäten durch einen programmierbaren Prioritätsencoder gelöst, der durch das Interrupt-Prioritätsregister R249 gesteuert wird. Der Ausgang des Prioritätsencoders zeigt zu dem Vektorplatz im Programmspeicher, der mit der Interruptanforderung, die zu bedienen ist, verbunden ist. Bevor die Inhalte des Interrupt-Maskenregisters (IMR) oder das Interrupt-Prioritätsregisters (IPR) geändert werden, muß das Interruptenablebit des IMR durch den Befehl "Disable Interrupt" (DI) zurückgesetzt werden. Die Anwendung des DI-Befehls ist unbedingt für die korrekte Interruptbehandlung erforderlich.

Name	Quelle	Vektorplatz	Bemerkung	
IRQ 0	$\overline{DAV0}$, RDY0, IRQ0	0, 1	ext. (P32)	H ↓ L flankengetriggert
IRQ 1	$\overline{DAV1}$, RDY1, IRQ1, BAK	2, 3	ext. (P33)	
IRQ 2	$\overline{DAV2}$, RDY2, IRQ2, T_{in}	4, 5	ext. (P31)	
IRQ 3	IRQ3	6, 7	ext. (P30)	
	Serieller Eingang	6, 7	int.	
IRQ 4	T0	8, 9	int.	
	Serieller Ausgang	8, 9	int.	
IRQ 5	T1	10, 11	int.	

Tabelle 3: Interrupttypen, -quellen und Vektorplätze

Falls eine Interruptanforderung gewährt wird, tritt der EMR in einen Interruptmaschinenzyklus ein, der alle folgenden Interrupts unwirksam macht, den Programmzähler und die Statusflags rettet und zu der Adresse springt, die der Vektorplatz für den Interrupt enthält. Erst an dieser Stelle geht die Steuerung zur Interruptserviceroutine über. Bild 27 stellt den Interruptzyklusprozess dar, falls eine Interruptanforderung eintrifft. Bild 18 stellt die aktuelle Interruptzeitfolge dar. Interrupts können wieder durch die Interruptbehandlungsroutine (EI-Befehl) aktiviert werden, um eine Interruptverschachtelung zu ermöglichen. Allerdings würde bei einer solchen Schachtelung der laufende Interrupt nicht mehr bei der Prioritätsermittlung berücksichtigt werden, da zu Beginn der Interruptserviceroutine bereits die Interruptanforderung im Interruptanforderungsregister gelöscht wird. (Wiederpriorisierte Interrupts können also (nach EI) die Serviceroutine des höherpriorisierten Interrupts unterbrechen). Interrupts können auch automatisch durch Verwenden eines Interruptreturnbefehls (IRET) als letzten Befehl der Interruptbehandlungsroutine wieder aktiviert werden. IRET speichert auch den Programmzähler und die Statusflags zurück.
Der EMR unterstützt beide Systeme, Abfrage- und Interruptsysteme. Zur Realisierung eines Abfragesystems können irgendwelche oder alle IRQ-Eingänge maskiert und das Interruptanforderungsregister abgefragt werden, um zu ermitteln, welche Interruptanforderung zu bedienen ist.

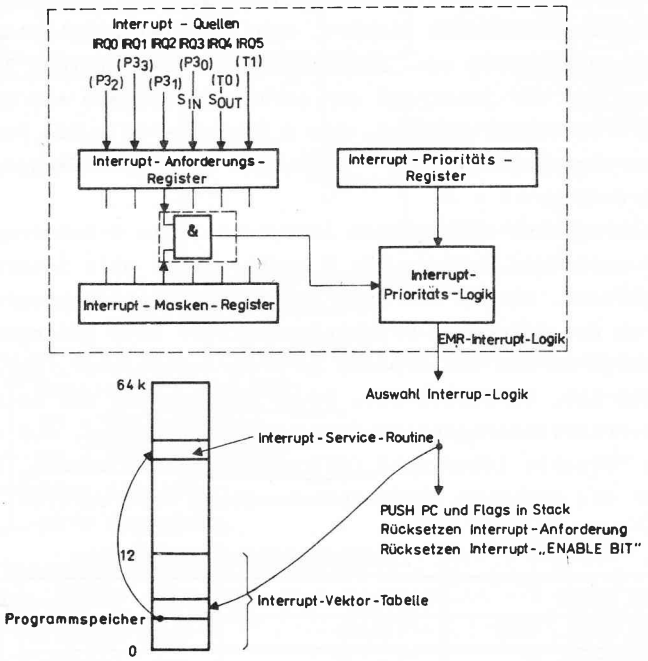


Bild 27: Interruptzyklusprozeß

3.10. Statusflags

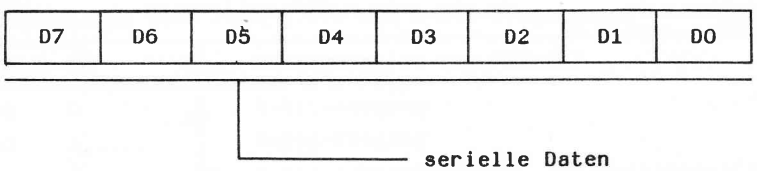
Das Flagregister R252 enthält 8 Flags:

C	Carry
Z	Zero
S	Sign
V	Overflow
D	Decimal Adjust
H	Half-Carry
F1	Anwenderflag 1
F2	Anwenderflag 2

Die Anwenderflags F1 und F2 können vom Anwender für allgemeine Zwecke genutzt werden. Die "Half-Carry"- und "Decimal Adjust"-Flags sind spezielle Flags, die nur für spezifische Befehle genutzt werden. Die übrigen Flags können vom Programmierer mit Sprung- und relativen Sprungbefehlen genutzt werden, um ein Repertoire von 19 Testbedingungen zu liefern. Im Abschnitt 6. wird gezeigt, wie die Flags durch den EMR-Befehlssatz beeinflußt werden. Die Flags können per Befehl gesetzt und rückgesetzt werden; jedoch nur jene Befehle, die die Flags nicht im Ergebnis der Ausführung beeinflussen, sollten verwendet werden (z. B. Lade Immediate). Zusätzlich kann das Carry-Flag durch den Set Carry Flag (SCF)-Befehl auf "1" gesetzt, durch den Reset Carry Flag (RCF)-Befehl gelöscht oder durch den Complement Carry Flag (CCF)-Befehl komplementiert werden.

4. Beschreibung der Steuerregister

4.1. R240 - Serielles Ein-/Ausgaberegister (SIO)

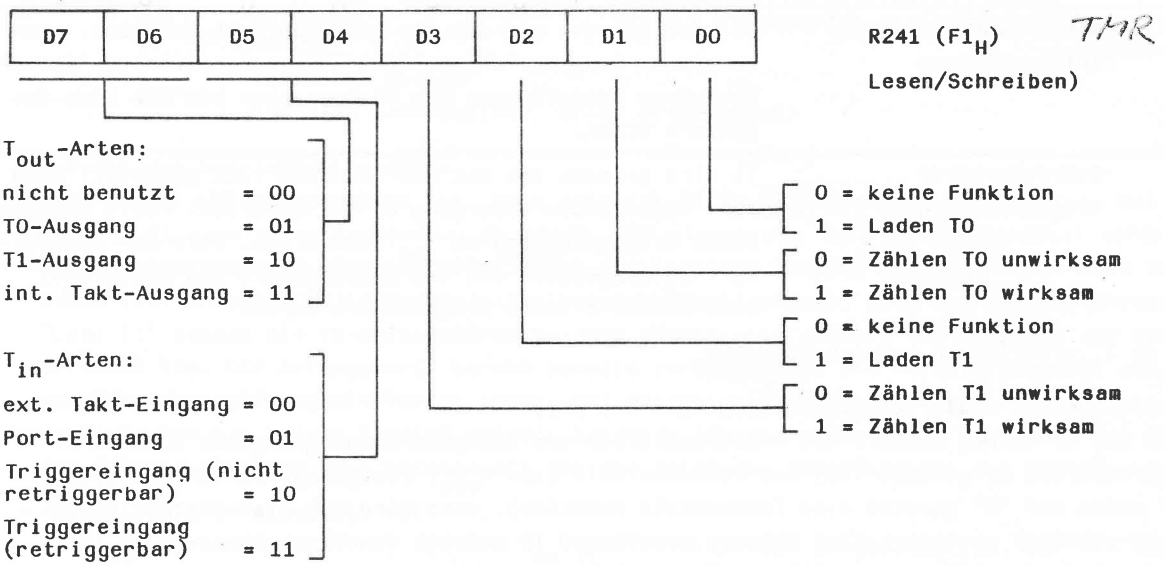


R240 (FO_H) SIO
(Lesen/Schreiben)

Lesen 240 = Empfangsdaten
Schreiben R240 = Sendedaten

Das Register R240 wird als serielles Ein-/Ausgabedatenregister verwendet, wenn Bit 6 des R247 auf 1 gesetzt ist. Damit werden P30 und P37 als serielle Ein- und Ausgangsleitungen konfiguriert. Wenn keine Parität gewünscht wird, enthält R240 auf allen 8 Bits Sendedaten. Wenn Parität gewählt wurde, enthält D7 von R240 die ungerade Parität während seriellem Sendens und ein Paritätsfehlerflog während des Empfangs. Parität wird durch Setzen von D7 des R247 ausgewählt. Falls R240 gelesen wird, wird das Zeichen in den Empfangspuffer gelesen; falls geschrieben wird, wird ein Zeichen in den Sender geladen.

4.2. R241 - Zeitgeberbetriebsartenregister (TMR)



Dieses Register wählt die Betriebsarten des Zähler-/Zeitgeber-Taktes aus und steuert TO und T1.

- Laden TO (DO):
  Nachdem dieses Bit gesetzt wurde, werden die Inhalte des TO-Laderegisters und des TO-Vorteiler-Laderegisters zum TO-Zähler und Vorteiler übertragen. Dadurch allein wird der Zähler jedoch nicht gestartet, DO wird jedoch nach dem Laden oder nach einem Masterreset automatisch zurückgesetzt.
- Zählen TO wirksam (D1):
  D1 aktiviert oder entaktiviert TO-Zähloperationen. Falls es gesetzt ist, werden die Werte in TO und seinem Vorteiler heruntergezählt, wobei der interne Takt benutzt wird. Falls es zurückgesetzt ist, wird das Herunterzählen unterbrochen. D1 wird nach einem Masterreset zurückgesetzt. Es ist zulässig, das Ladebit DO und das Zählaktivierungsbit D1 gleichzeitig zu setzen.
- Laden T1 (D2):
  Funktion analog "Laden TO"
- Zählen T1 wirksam (D3):
  Funktion analog "Zählen TO wirksam"
- Betriebsarten des externen Zeitgebereinganges (D4, D5):
  Über D4 und D5 werden codiert die 4 Betriebsarten des externen Zeitgebereinganges (T_{in}) für den Zähler/Zeitgeber definiert. Vorher muß jedoch P31 als ein externer Zeitgebereingang definiert werden (R243, D1)!

D5	D4	T_{in} benutzt als	Bemerkung
0	0	ext. Takteingang	T_{in} wird benutzt als ein externer Takt für T1. In dieser Betriebsart umgeht der externe Takt den 1 : 4-Teiler und steuert direkt den Vorteiler an.
0	1	Toreingang für internen Takt (aktiv "High")	T1 wird durch internen Takt getaktet (XTAL-Frequenz geteilt durch 8), der durch diesen Eingang getort wird. Falls der Toreingang von "High" auf "Low" schaltet, wird falls freigegeben, ein Interrupt erzeugt.
1	0	Nicht retriggerbarer Triggereingang	T1 wird geladen und mit dem internen Takt getaktet, wenn an diesem Eingang "High-to-Low"-Übergang auftritt. Weitere Übergänge beeinflussen die T1-Operation bis zum Ende des Zählers nicht.
1	1	Retriggerbarer Triggereingang	T1 wird geladen und mit dem internen Takt getaktet, wenn ein "High-to-Low"-Übergang an T_{in} auftritt. Falls zusätzliche Triggerimpulse an T_{in} eintreffen, wird der Anfangswert zurückgeladen und die Zählung wird neu gestartet, wenn modulo-n-count programmiert wurde.

Tabelle 4: T_{in}-Funktionen und Arbeitsweise

Betriebsarten des Zählers/Zeitgeberausganges D6, D7):
Über D6 und D7 werden codiert die 4 Betriebsarten des Zeitgeberausgangssignals definiert. Wenn D6 oder/und D7 gesetzt ist, muß P36 ebenfalls (als T_{out}) festgelegt werden. Falls D6 und D7 beide auf "0" gesetzt sind (unbenutzte Funktion), dann wird P36 als Ausgabebit und durch D5 von R247 gesteuert. Bei Nutzung von T0 und T1 erfolgt das Umschalten von T_{out} durch das Ende der Zählung von T0 oder T1.

D7	D6	T_{out}-Funktion
0	0	nicht benutzt
0	1	T0-Ausgang
1	0	T1-Ausgang
1	1	Systemtakt (1/2 XTAL-Frequenz)

4.3. R242 - Zähler-/Zeitgeberregister 1 (T1)

D7	D6	D5	D4	D3	D2	D1	D0

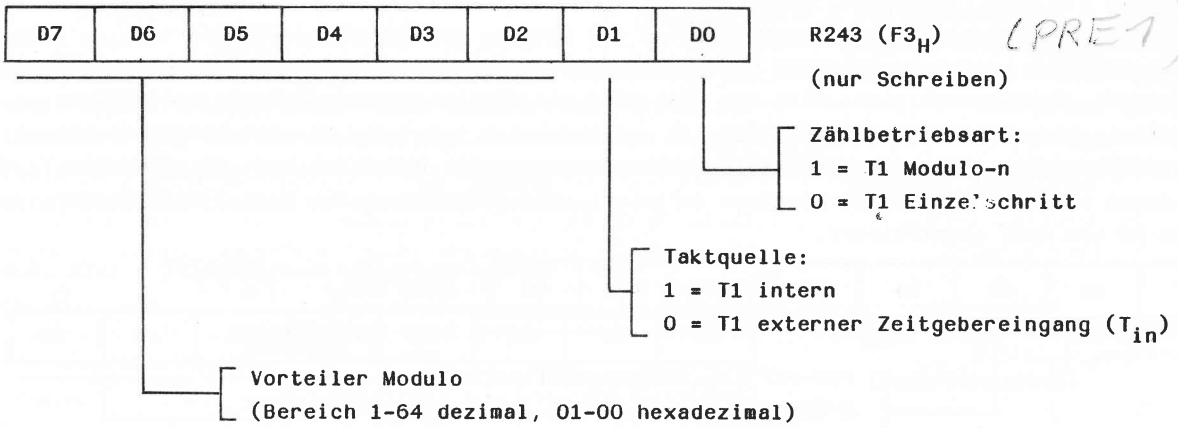
R242 ($F2_H$)
(Lesen/Schreiben)

wenn geschrieben:
= T1-Anfangswert (Bereich 1-256 dezimal, 01-00 hexadezimal)

wenn gelesen:
= aktueller Zählerstand von T1

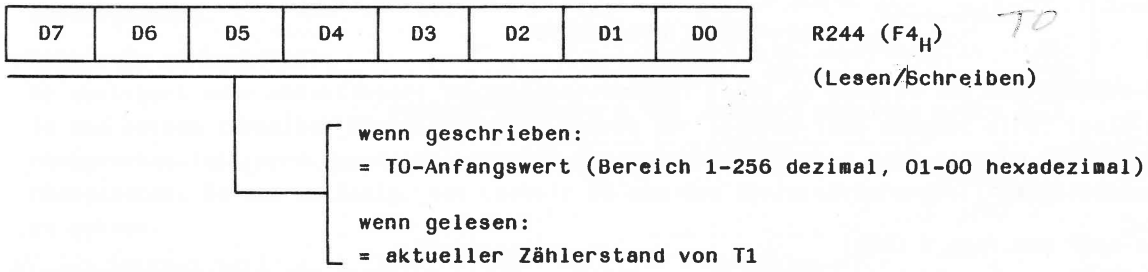
4.4. R243 - Vorteilerladeregister 1 (PRE1)

Dieses Register speichert den T1-Vorteiler-Anfangswert und definiert die T1-Taktquellen und Zählbetriebsarten.



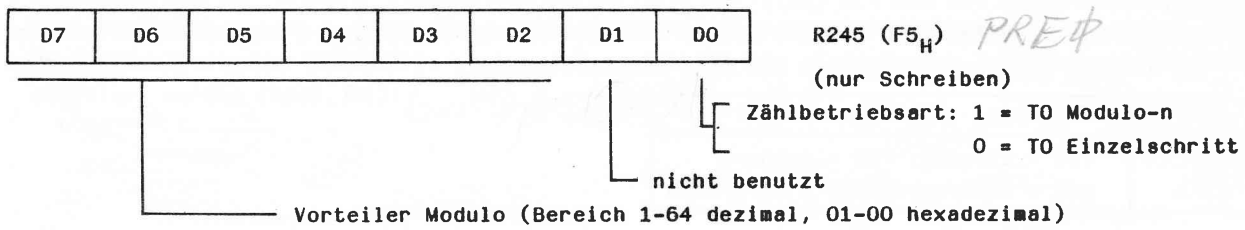
- Auswahl der Betriebsarten (D0):
  Wenn dieses Bit zurückgesetzt ist, dann arbeitet T1 im Einzelschritt-Zählbetrieb, bei dem der Wert in T1 nach jedem Ladebefehl T1 (R241, D2) einmal bis Null heruntergezählt wird. Eine Interruptanforderung wird ausgelöst, wenn das Ende der Zählung erreicht wird. Wenn D0 gesetzt ist, arbeitet T1 im Modulo-n-Zählbetrieb (fortlaufend). Nach Empfang des Befehls "Lade T1" werden die T1-Anfangswerte geladen und heruntergezählt bis das Ende der Zählung erreicht ist. Die Anfangswerte werden jeweils zurückgeladen und heruntergezählt, solange das Zähleraktivierungsbit für T1 (R241, D3) gesetzt ist. Das Laden des Laderegisters mit neuen Werten hat keinen Einfluß auf die laufende Zähloperation. Beim Erreichen des Endes der Zählung wird der neue Anfangswert für die folgenden Zählzyklen in den Zähler geladen.
- Auswahl der T1-Taktquelle (D1):
  Falls D1 zurückgesetzt ist, liefert T_{in} den T1-Takt. Wenn es gesetzt ist, liefert der interne Takt (Systemtakt geteilt durch 4) den T1-Takt. Bei Benutzung von T_{in} für den Takt muß im TMR-Register (R241) die entsprechende Betriebsart festgelegt werden!
- Vorteilerwert von T1 (D2-D7):
  Dieser 6-Bit-Wert bestimmt den Modulo des Vorteilers. Das niedrigste Bit ist D2.

4.5. R244 - Zähler-/Zeitgeberregister 0 (T0)



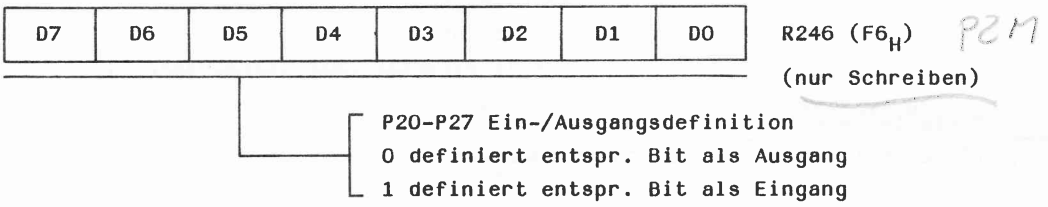
4.6. R245 - Vorteilerladeregister 0 (PRE0)

Dieses Register hat dieselben Funktionen wie das entsprechende Register von T1 (R243), mit Ausnahme von D1, das nicht benutzt wird.



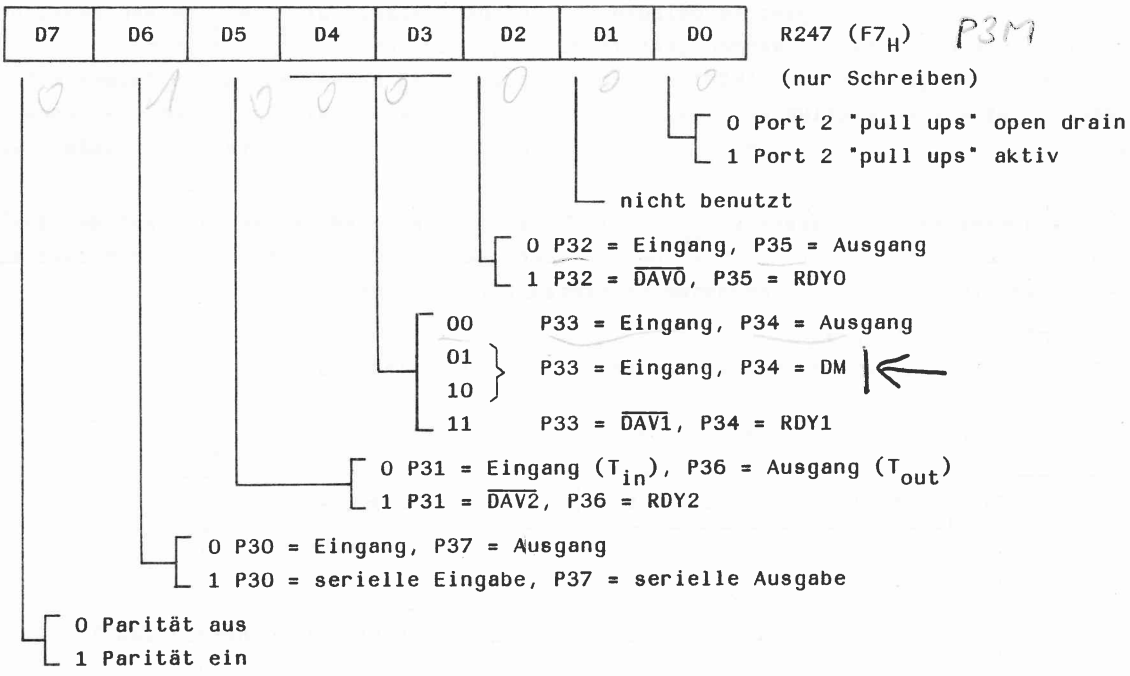
4.7. R246 - Betriebsart Port 2 (P2M)

Dieses Register kann jedes Bit von Port 2 als eine Eingangs- oder eine Ausgangsleitung programmieren. Beim Setzen eines Bits von R246 wird die damit korrespondierende Leitung von Port 2 als Eingang definiert. Wenn ein Bit zurückgesetzt ist, wird die entsprechende Leitung als Ausgang definiert. Nach einem Master-Reset enthält R 246 den Wert FF_H und alle Leitungen von Port 2 sind als Eingänge definiert. Die Betriebsart von Port 2 wird ferner durch D0 von R247 spezifiziert.



4.8. R247 - Betriebsart Port 3 (P3M)

Dieses Register legt fest, welche Leitungen von Port 3 für parallele Ein-/Ausgabe, Interruptanforderung, Zeitgeber-Ein-/Ausgabe und Handshake- oder Statusausgänge verwendet werden.



- "Pull-ups" von Port 2 (D0):
  Im zurückgesetzten Zustand liefert dieses Bit Open-Drain-Ausgänge für Port 2, indem die aktiven "Pull-Ups" in ihrer Wirkung aufgehoben werden. Open-Drain-Ausgänge erlauben, daß die Leitungen von Port 2 mit anderen Signalen Wired-Or verknüpft werden können.
- Nicht benutzt (D1)
- Betriebsart von P32 und P35 (D2):
  Dieses Bit legt fest, ob P32 und P35 für die Ein-/Ausgabe von Port 3 verwendet werden oder ob sie als Handshake-Leitungen das Port 0 unterstützen.

D2	Funktion
0	P32 = Eingang, P35 = Ausgang
1	P32 = $\overline{DAV0}$/RDY0, P35 = RDY0/$\overline{DAV0}$

• Betriebsart von P33 und P34 (D3, D4):
Diese Bits legen fest, ob P33 und P34 für die Ein-/Ausgabe verwendet werden oder ob sie die Funktionen von Port 1 unterstützen.

D4 und D3	Funktion
0 0	P33 = Eingang, P34 = Ausgang
0 1	P33 = Eingang, P34 = $\overline{DM}$
1 0	
1 1	P33 = $\overline{DAV1}$/RDY1, P34 = RDY1/$\overline{DAV1}$

D4 D3
we : 0 1

• Betriebsart von P31 und P36 (D5):
Dieses Bit legt fest, ob P31 und P36 für Ein-/Ausgabe (D5=0) konfiguriert werden oder zur Handshake-Steuerung (D5=1) als Unterstützung von Port 2 benutzt werden. Wenn sie für Ein-/Ausgabe programmiert sind, spiegeln P31 und P36 gewöhnlich die R3-Registerbits 1 und 6 wieder, es können jedoch andere Datenquellen wie folgt ausgewählt werden:
Wenn Bit D1 vom T1-Vorteilerregister R243 zurückgesetzt wird, wird P31 der Zeitgebersteuereingang (T_{in}) und verhält sich entsprechend der Wahrheitstabelle, die die Bits D4 und D5 des Zeitgeberbetriebsartenregisters R241 beschreibt. Bit D1 von R243 muß gesetzt werden, um P31 als eine Dateneingangsleitung für R3 zu aktivieren. Der Ausgang P36 wird T_{out} (entweder für TO oder für T1) oder SCLK in Abhängigkeit der Bits D6 und D7 des Betriebsartenregisters des Zeitgebers (R241). Um P36 als Datenausgang für R3 zu aktivieren, müssen die Bits D6 und D7 beide auf Null zurückgesetzt werden. Wenn P31 und P36 zur Handshake-Steuerung, haben D4-D7 von R241 und D1 von R243 keinen Einfluß.

D5	Funktion
0	P31 = Eingang (T_{in}), P36 = Ausgang (T_{out})
1	P31 = $\overline{DAV2}$/RDY2, P36 = RDY2/$\overline{DAV2}$

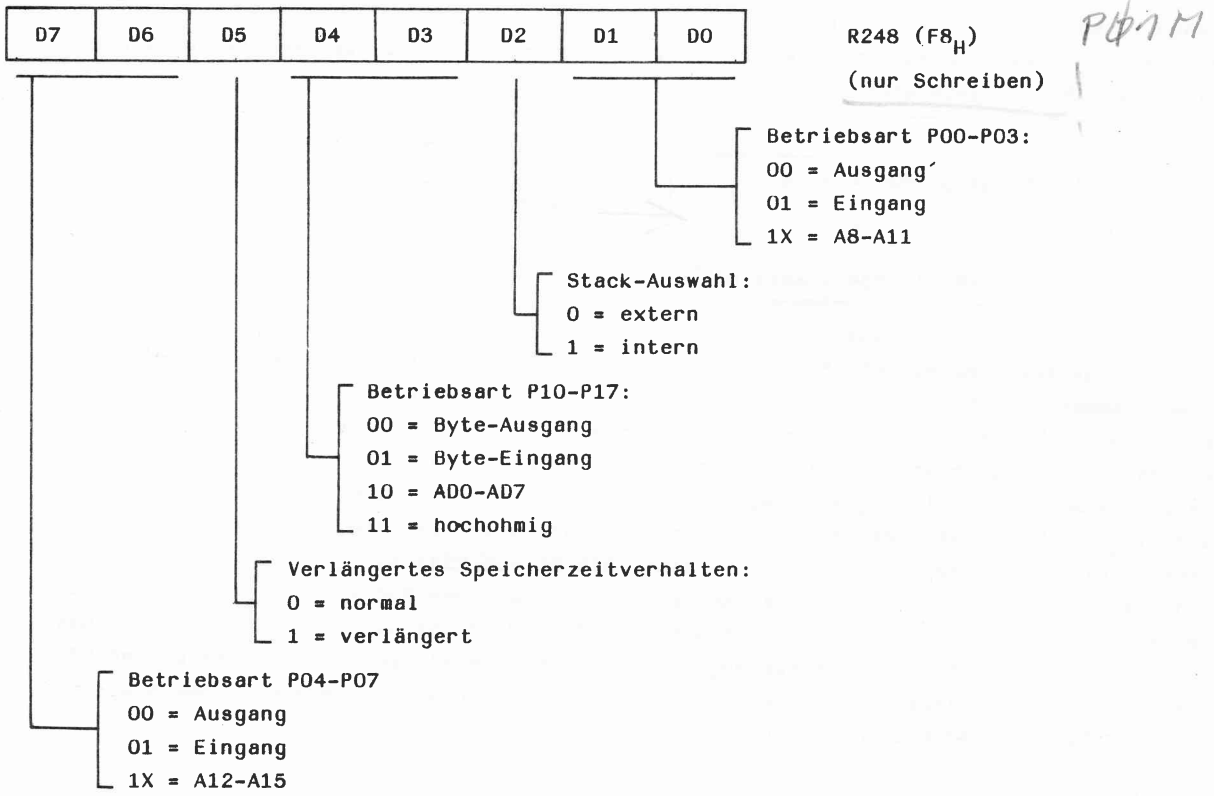
• Betriebsart von P30 und P37 (D6):
Dieses Bit legt fest, ob P30 und P37 für Ein-/Ausgabe oder für serielles Empfänger-/Sender-Interface benutzt werden.

D6	Funktion
0	P30 = Eingang, P37 =Ausgang
1	P30 = serielle Eingabe, P37 = serielle Ausgabe

• Parität D7):
Wenn mittels D6 serielle Ein-/Ausgabe ausgewählt wird, dann erhält man durch Setzen von D7 ungerade Parität für die Sendedaten und ungerade Parität bei der Prüfung der Empfangsdaten.

4.9. R248 - Betriebsarten Ports 0 und 1 (P01M)

Dieses Register konfiguriert die Ports 0 und 1, wählt einen internen oder externen Stack und wählt normales Speicherzeitverhalten aus.



- Betriebsarten von Port 0 (D0, D1):
  Mit den Bits D0 und D1 von R248 wird die Betriebsart der Leitungen P00-P03 von Port 0 festgelegt. Nach einem Rücksetzen wird das untere Nibbel von Port 0 für die Eingabebetriebsart konfiguriert. Vorrangig ist jedoch D7 zu beachten:
  Wenn D7 von R248 gesetzt ist, dann enthält das untere Nibbel von Port 0 immer A8-A11, unabhängig von der durch D0 und D1 vorgegebenen Konfiguration!

D7	D1	D0	Konfiguration von P00-P03
0	0	0	4-Bit-Ausgang
0	0	1	4-Bit-Eingang
0	1	X	4-Bit-Adresse (A8-A11)
1	X	X	4-Bit-Adresse (A8-A11)

- Auswahl des Stack-Bereiches (D2):
  Wenn D2 gesetzt ist, befindet sich der Stack in der internen Registerdatei und durch den Stack-Pointer (Kellerspeicherzeiger) SPL (R255) wird auf ihn gezeigt. Wenn D2 zurückgesetzt ist, befindet sich der Stack im externen Datenspeicher und durch den Stack-Pointer SP (R254 und R255) wird auf ihn gezeigt. Während der Nutzung des internen Stacks (D2=1) kann R254 als ein Datenregister verwendet; der Über- oder Unterlauf von R255 muß dann jedoch exakt gehandelt werden.

- Betriebsart von Port 1 (D3, D4):
  Mit diesen Bit wird die Betriebsart der Leitungen P10-P17 von Port 1 festgelegt. Nach einem Rücksetzen wird Port 1 als die 8-Bit-Eingangsport konfiguriert.

D4	D3	Konfiguration von P10-P17
0	0	8-Bit-Ausgang
0	1	8-Bit-Eingang
1	0	8-Bit-Adressen/Daten, zeitmultiplexe(AD0-AD7)
1	1	hochohmiger Zustand

In der nachstehenden Tabelle wird der Zusammenhang zwischen dem hochohmigen Zustand und den durch R248 ausgewählten Adreß-Betriebsarten verdeutlicht:

D7	D4	D3	D1	3-State-Leitungen
0	1	1	0	Port 1, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$
0	1	1	1	Port 1, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, P00-P03
1	1	1	X	Port 1, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, P00-P07

- Verlängertes Speicherzeitverhalten (D5):
  Wenn dieses Bit gesetzt ist, wird der Speicherzyklus um eine Taktperiode verlängert, was den Anschluß langsamerer Speicher ermöglicht. Wenn es rückgesetzt ist, wird normales Zeitverhalten für externe Speicher ausgewählt.

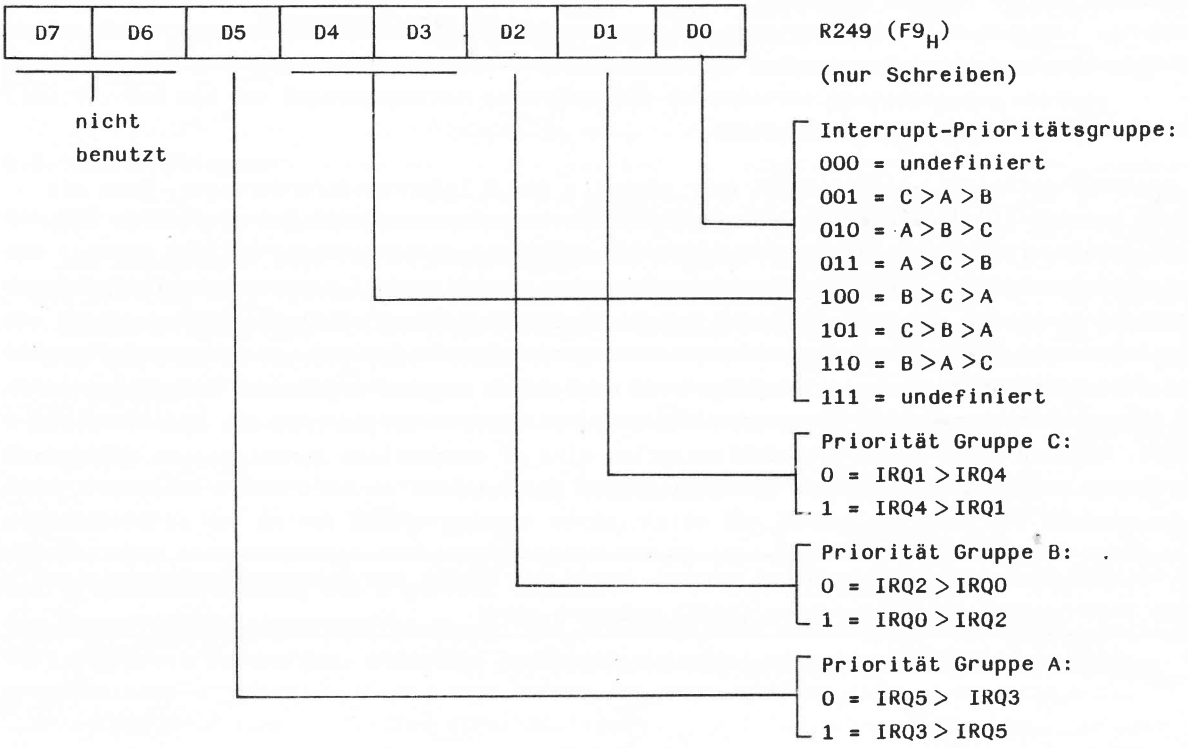
- Betriebsart von Port 0 (D6, D7):
  Diese Bits steuern die Betriebsart der Leitungen P04-P07 von Port 0. Nach einem Rücksetzen wird das obere Nibbel von Port 0 für die Betriebsart "Eingabe" konfiguriert. Wenn D7 gesetzt ist, werden P00-P03 zu A8-A11 - unabhängig davon, ob die Bits D0 und D1 eine andere Konfiguration festlegen.

D7	D6	Konfiguration von P04-P07
0	0	4-Bit-Ausgang
0	1	4-Bit-Eingang
1	X	4-Bit-Adresse (A12-A15)

4.10. R249 - Interruptprioritätsregister (IPR)

Dieses Register priorisiert die 6 Ebenen der vektorisierten Interrupts. Es können 48 verschiedene Reihenfolgen priorisiert werden, um gleichzeitige Interruptanforderungen entflechten zu können. Die 6 Interruptebenen IRQ0-IRQ5 werden in 3 Gruppen unterteilt - hier mit A, B und C bezeichnet - die je 2 Interruptanforderungen enthalten:

A: IRQ3 (SI/P30) und IRQ5 (T1)
B: IRQ0 (P32) und IRQ2 (P31)
C: IRQ1 (P33) und IRQ4 (SO/T0)



• Die Bits D1, D2 und D5 von R249 bestimmen die Priorität der individuellen Komponenten innerhalb der einzelnen Gruppen:

Gruppe	Bit	Priorität höchste	niedrigste
C	D1 = 0	IRQ1	IRQ4
	1	IRQ4	IRQ1
B	D2 = 0	IRQ2	IRQ0
	1	IRQ0	IRQ2
A	D5 = 0	IRQ5	IRQ3
	1	IRQ3	IRQ5

• Die Bits D0, D3 und D4 von R249 legen die Priorität der Gruppen A, B und C untereinander fest:

D4	D3	D0	höchste Priorität →	niedrigste
0	0	0	nicht benutzt	
0	0	1	C	A B
0	1	0	A	B C
0	1	1	A	C B
1	0	0	B	C A
1	0	1	C	B A
1	1	0	B	A C
1	1	1	nicht benutzt	

4.11. R250 - Interruptanforderungsregister (IRQ)

Dieses Register speichert die Interruptanforderungen. Da es ein Lese-/Schreibregister ist, kann es auch für Abfrage (Polling) verwendet werden. Bei einem Interrupt von einer der 6 Ebenen wird in das entsprechende Bit eine "1" eingeschrieben. Die Bits D0-D5 von R250 entsprechen genau den Interruptanforderungen IRQ0-IRQ5. D6 und D7 werden nicht benutzt.

D7	D6	D5	D4	D3	D2	D1	D0
nicht benutzt		IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0

R250 (FA_H) IRQ

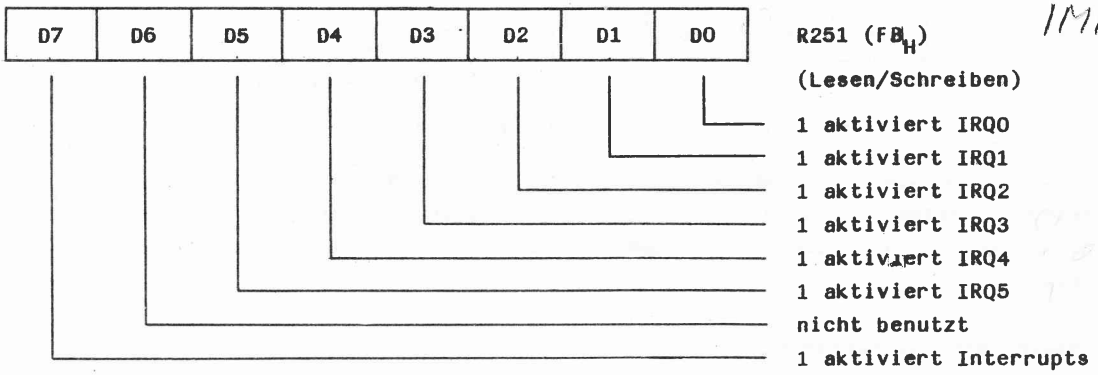
(Lesen/Schreiben)

4.12. R251 - Interruptmaskenregister (IMR)

Dieses Register aktiviert - individuell oder global - die 6 Interruptanforderungen. Wenn die Bits D0-D5 gesetzt sind, dann sind die entsprechenden Interruptanforderungen aktiviert. Bevor jedoch irgendeine (individuelle) Interruptanforderung anerkannt werden kann muß D7, das "Master"-Aktivierungsbit gesetzt werden. Das Rücksetzen von D7 macht global alle Interruptanforderungen unwirksam. D7 wird durch den Befehl "Enable Interrupt" (EI) gesetzt und durch "Disable Interrupt" (DI) rückgesetzt. D7 kann auch durch Ladebefehle gesetzt werden, hat aber nur dann eine interruptfreigebende Wirkung, wenn nach Reset bereits irgendwann einmal der Befehl EI abgearbeitet wurde. Ein Rücksetzen von D7 durch Ladebefehle bewirkt ein Sperren aller Interrupts. Während eines Interruptmaschinenzyklus wird D7 automatisch zurückgesetzt und nach der Ausführung des "Interrupt-Return"-Befehls (IRET) gesetzt.

Achtung!

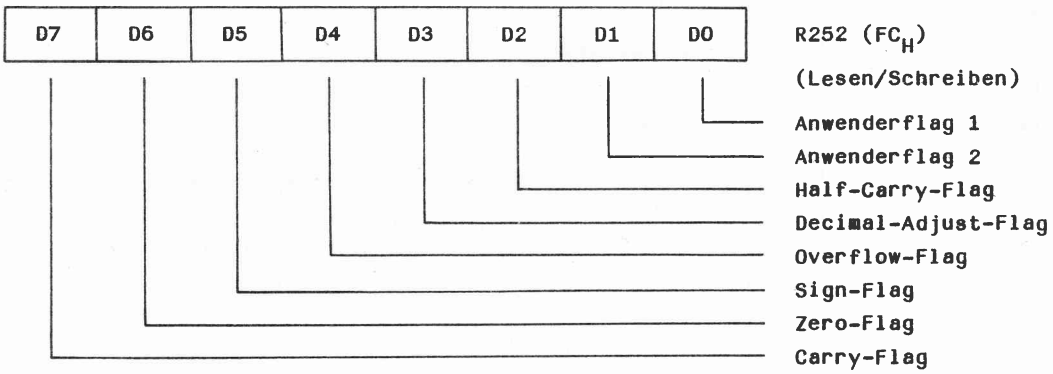
D7 muß mittels DI-Befehl zurückgesetzt werden, bevor der Inhalt des Interruptmaskenregisters oder des Interruptprioritätsregisters (R249) geändert wird!



IMR

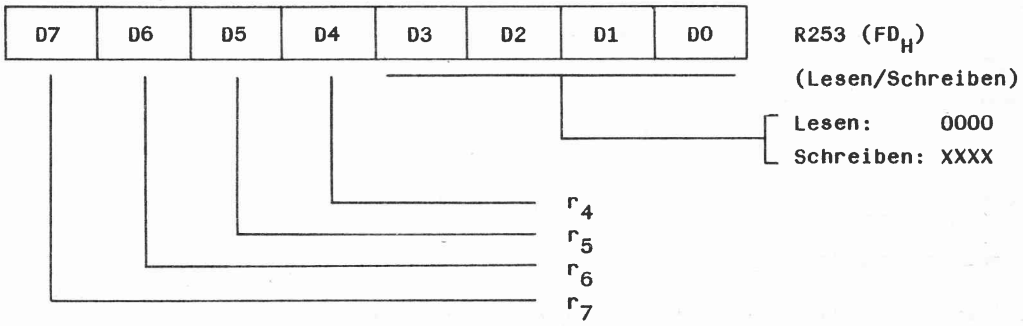
4.13. R252 - Flagregister (FLAGS)

Die Bits D2-D7 enthalten die Statusflags, die Bits D0 und D1 sind vom Anwender definierbar.



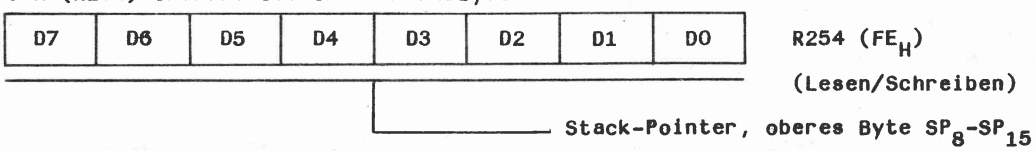
4.14. R253 - Register-Pointer (RP)

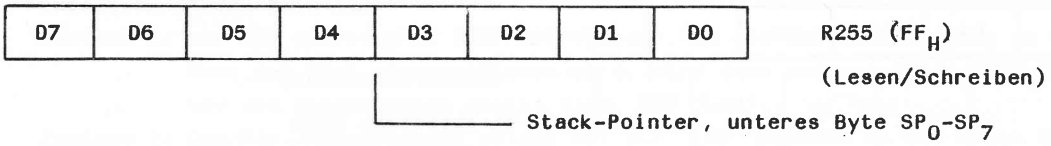
Die vier oberen Bits dieses Registers bilden einen Registerzeiger, der zum Anfang der laufenden Arbeitsregistergruppe zeigt. Die unteren vier Bits (Registerkennzeichner), die ein Register innerhalb der Arbeitsregistergruppe spezifizieren, werden durch den entsprechenden Befehl geliefert. Beim Lesen des Register-Pointers sind die 4 niederwertigen Bits (D0-D3) immer Null, beim Schreiben können sie beliebig sein.



4.15. R254, R255 - Stack-Pointer (SPH, SPL)

Der Stack-Pointer (Kellerspeicherzeiger) setzt sich aus zwei 8-Bit-Registern zusammen:
SPL (R255) enthält das niedere Adreßbyte
SPH (R254) enthält das obere Adreßbyte





Bekanntlich legt Bit D2 von R248 fest, ob der Stack in der Registerdatei oder im externen Speicher liegt. Wenn der Stack intern ist, wird R254 nicht als Stack-Pointer benutzt und ist als Datenregister verwendbar. Dabei muß jedoch der Über- oder Unterlauf von R255 exakt gehandhabt werden.

4.16. Registerinhalte nach dem Rücksetzen

Tabelle 4.2. gibt Auskunft über die Inhalte der Steuerregister R240-R255 nach einem Rücksetzen.

Steuerregister			D7	D6	D5	D4	D3	D2	D1	D0	Bemerkung
	Funktion	Adr. Hex									
R240	SIO	F0	nicht definiert								-
R241	TMR	F1	0	0	0	0	0	0	0	0	Stoppt T0 und T1
R242	T1	F2	nicht definiert								-
R243	PRE1	F3	X	X	X	X	X	X	0	0	Einzelschritt-Zählbetrieb, externe Taktquelle
R244	T0	F4	nicht definiert								-
R245	PRE0	F5	X	X	X	X	X	X	X	0	Einzelschritt-Zählbetrieb
R246	P2M	F6	1	1	1	1	1	1	1	1	Port 2 = Eingänge
R247	P3M	F7	0	0	0	0	0	0	X	0	Port 2 = "Pull ups" open drain, P30-P33 = Eingänge, P34-P37 = Ausgänge
R248	P01M	F8 =4DH	0	1	0	0	1	1	0	1	Ports 0, 1 = Eingänge, normaler Speicherzyklus, Stack intern
R249	IPR	F9	nicht definiert								-
R250	IRQ	FA	X	X	0	0	0	0	0	0	Interruptanforderungen zurückgesetzt
R251	IMR	FB	0	X	X	X	X	X	X	X	Interrupts unwirksam
R252	FLAGS	FC	nicht definiert								-
R253	RP	FD	nicht definiert								-
R254	SPH	FE	nicht definiert								-
R255	SPL	FF	nicht definiert								-

Tabelle 4: Registerinhalte nach dem Rücksetzen

5. Befehlssatz der EMR-Schaltkreise

5.1. Funktionelle Zusammenfassung der Befehle

Sämtliche EMR-Befehle lassen sich funktionell in 8 Gruppen unterteilen:

- Ladebefehle
- Arithmetik-Befehle
- Logische Befehle
- Programmsteuerbefehle (Sprungbefehle)
- Bitbehandlungsbefehle (Testbefehle)
- Blocktransferbefehle
- Rotations- und Schiebebefehle
- CPU-Steuerbefehle

Im folgenden werden die einzelnen Befehle (gruppenweise) zusammengefaßt dargestellt (siehe auch 5.3.).

• Ladebefehle

Befehl	Operand (en)	Bedeutung
CLR	dst	Clear
LD	dst, src	Load
LDC	dst, src	Load Constant
LDE	dst, src	Load External Data
POP	dst	Pop
PUSH	src	Push

• Arithmetikbefehle

Befehl	Operand (en)	Bedeutung
ADC	dst, src	Add With Carry
ADD	dst, src	Add
CP	dst, src	Compare
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
SBC	dst, src	Subtract With Carry
SUB	dst, src	Subtract

• Logische Befehle

Befehl	Operand (en)	Bedeutung
AND	dst, src	Logical And
COM	dst	Complement
OR	dst, src	Logical OR
XOR	dst, src	Logical Exclusive OR

• Programmsteuerbefehle

Befehl	Operand (en)	Bedeutung
CALL	dst	Call
DJNZ	r, dst	Decrement and Jump If Nonzero
IRET		Interrupt Return
JP	cc, dst	Jump
JR	cc, dst	Jump Relative
RET		Return

• Bitbehandlungsbefehle

Befehl	Operand (en)	Bedeutung
TCM	dst, src	Test Complement Under Mask
TM	dst, src	Test Under Mask
AND	dst, src	Logical And
OR	dst, src	Logical OR
XOR	dst, src	Logical Exclusive OR

• Blocktransferbefehle

Befehl	Operand (en)	Bedeutung
LDCI	dst, src	Load Constant Autoincrement
LDEI	dst, src	Load External Data Autoincrement

- **Rotations- und Schiebebefehle**

Befehl	Operand (en)	Bedeutung
RL	dst	Rotate Left
RLC	dst	Rotate Left Through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right Through Carry
SRA	dst	Shift Right Arithmetik
SWAP	dst	Swap Nibbels

- CPU-Steuerbefehle

Befehl	Operand	Bedeutung
CCF		Complement Carry Flag
DI		Disable Interrupts
EI		Enable Interrupts
NOP		No Operation
RCF		Reset Carry Flag
SCF		Set Carry Flag
SRP	src	Set Register Pointer

5.2. Flags und Bedingungscodes

Zur Programmsteuerung des EMR sind im Steuerregister R252 folgende 6 Flags vorgesehen:

Kurzbezeichnung	Bedeutung
C	Carry-Flag
Z	Zero-Flag
S	Sign-Flag
V	Overflow-Flag
D	Decimal-Adjust-Flag
H	Half-Carry-Flag

- Carry-Flag:
  Das Carry-Flag wird bei folgenden Befehlen wirksam:
  Addition (ADD, ADC), Subtraktion (SUB, SBC), Compare (CP), Decimal Adjust (DA), Rotate (RL, RLC, RR, RRC), Swap (SWAP) und Interrupt Return (IRET). Wenn es gesetzt ist, kennzeichnet das Carry-Flag generell einen Übertrag vom Bit 7 eines als Akkumulator verwendeten Registers.

- Zero-Flag:
  Das Zero-Flag wird bei denselben Befehlen wirksam wie beim Carry-Flag, zusätzlich bei den Logikbefehlen (AND, COM, OR, XOR), bei Increment- und Decrementbefehlen (INC, INCW, DEC, DECW) und bei Bitbehandlungsbefehlen (TCM, TM). Das Zero-Flag wird gesetzt, wenn das Ergebnis einer Operation gleich Null ist.

- Sign-Flag:
  Das Sign-Flag wird bei denselben Befehlen wirksam wie beim Zero-Flag. Das Sign-Flag wird gesetzt, wenn Bit 7 des Ergebnisses einer Operation gleich "1" (Vorzeichen minus) ist.

- Overflow-Flag:
  Das Overflow-Flag wird bei denselben Befehlen wirksam wie beim Zero- und Sign-Flag. Wenn es gesetzt ist, kennzeichnet das Overflow-Flag, daß ein fehlerhaftes Zweikomplement-Ergebnis vorliegt, da es den Bereich von -128 bis +127 überschritten hat, der im Zweierkomplement dargestellt werden kann.

- Decimal-Adjust-Flag:
  Das Decimal-Adjust-Flag wird für die BCD-Arithmetik benutzt und zur Wandlung von 8-Bit-Binärzahlen in zwei 4-Bit-BCD-Digits. Da der Algorithmus für die Korrektur von BCD-Operationen für Addition und Subtraktion unterschiedlich ist, spezifiziert dieses Flag den zuletzt

ausgeführten Befehl. Das Decimal-Adjust-Flag wird bei der Ausführung von Subtraktionsbefehlen (SUB, SBC) auf "1" gesetzt und bei Additionsbefehlen (ADD, ADC) auf "0" zurückgesetzt.

- Half-Carry-Flag:
  Das Half-Carry-Flag signalisiert einen Übertrag von Bit 3 eines als Akkumulator benutzten Registers. Dieses Flag wird als Ergebnis von ausgeführten Subtraktions- (SUB, SBC) oder Additionsbefehlen (ADD, ADC) wirksam.

- Bedienungscodes:
  Bei direkten und relativen Sprungbefehlen (JP, JR) wird zur Spezifizierung der Testbedingung ein 4-Bit-Bedingungsfeld ("cc") benutzt. Diese Bedingungscodes gehen als "variable" Komponente in das erste Byte des Sprungbefehls von 2 bzw. 3 Byte Länge ein.

Bedingungscode	Hex (Oberes Nibbel)	Bedeutung	gesetzte Flags
	8	immer wahr	———
C	7	"Carry"	C=1
NC	F	kein "Carry"	C=0
Z	6	Null	Z=1
NZ	E	nicht Null	Z=0
PL	D	plus	S=0
MI	5	minus	S=1
OV	4	"Overflow"	V=0
NOV	C	kein "Overflow"	V=1
EQ	6	gleich	Z=1
NE	E	nicht gleich	Z=0
GE	9	größer als oder gleich	(S XOR V)=0
LT	1	kleiner als	(S XOR V)=1
GT	A	größer als	(Z OR (S XOR V))=0
LE	2	kleiner als oder gleich	(Z OR (S XOR V))=1
UGE	F	vorzeichenfrei größer als oder gleich	C=0
ULT	7	vorzeichenfrei kleiner als	C=1
UGT	B	vorzeichenfrei größer als	(C=0 AND Z=0)=1
ULE	3	vorzeichenfrei kleiner als oder gleich	(C OR Z)=1
	Ø	niemals wahr	———

5.3. Notierung

Zur Beschreibung der Adressierungsarten und Befehlsoperationen wird folgende Notierung benutzt (siehe auch 5.4.):

- Adressierungsarten:

Symbol	Bedeutung
R	Register- oder Arbeitsregisteradresse
r	nur Arbeitsregisteradresse
IR	indirekte Register- oder indirekte Arbeitsregisteradresse
Ir	nur indirekte Arbeitsregisteradresse
RR	Registerpaar- oder Arbeitsregisterpaar-Adresse
IRR	indirekte Registerpaar- oder Arbeitsregisterpaar-Adresse
Irr	nur indirektes Arbeitsregisterpaar
X	indizierte Adresse
DA	direkte Adresse
RA	relative Adresse
IM	unmittelbare Adresse

- zusätzliche Symbole:

Symbol	Bedeutung
dst	Zielspeicheradresse oder -inhalt
src	Quellspeicheradresse oder -inhalt
cc	Bedingungscode (siehe 5.2.)
ⓐ	indirekter Adresspräfix
SP	Stack Pointer/Kellerspeicherzeiger (Steuerregister R254, 255)
PC	Programmzähler
FLAGS	Flagregister (Steuerregister R252)
RP	Registerzeiger (Steuerregister R253)
IMR	Interruptmaskenregister (Steuerregister R251)

- Flags, die von einem Befehl beeinflußt werden, sind gekennzeichnet durch:

  0 = gelöscht zu Null
  1 = gesetzt auf Eins
  ↕ = entsprechend der Operation gelöscht oder gesetzt
  • = unbeeinflußt
  X = undefiniert

- Die Zuweisung eines Wertes wird durch das Symbol "←" gekennzeichnet, z. B.

  dst ← dst + src

  (Quelldaten werden zu den Zieldaten addiert und das Ergebnis in der Zielspeicherzelle gespeichert.)

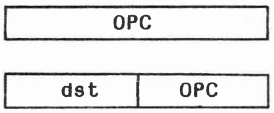
- Die Notierung "addr (n)" wird benutzt, um sich auf das Bit n einer gegebenen Speicherzelle zu beziehen, z. B.

  dst (7)

  (Operation bezieht sich auf Bit 7 des Zieloperanden.)

5.4. Zusammenfassung der Befehle

In der am Ende dieses Abschnitts folgenden Tabelle werden die EMR-Befehle einschließlich der anwendbaren Adressierungsarten definiert. Es sind zusätzlich die Opcodes, die Anzahl der Bytes und der internen Taktzyklen und die jeweils beeinflußten Flags aufgelistet. Die Anzahl der internen Taktzyklen, die in der Spalte "Ausführung/Zyklen" aufgelistet sind, stellen die Werte dar, die zu nutzen sind, um die Ausführungszeit eines Programmes zu berechnen. Die Zeit vom Beginn eines Befehls bis zur Datenänderung ist die Summe der "Ausführungszyklen" und der "Pipeline-Zyklen". Der Beginn eines Befehls wird als der erste interne Takt definiert, der einem Befehls-"Sync" folgt.



CCF, DI, EI, IRET, NOP, RCF, RET, SCF

INC r

Bild 28: Formate von Ein-Byte-Befehlen

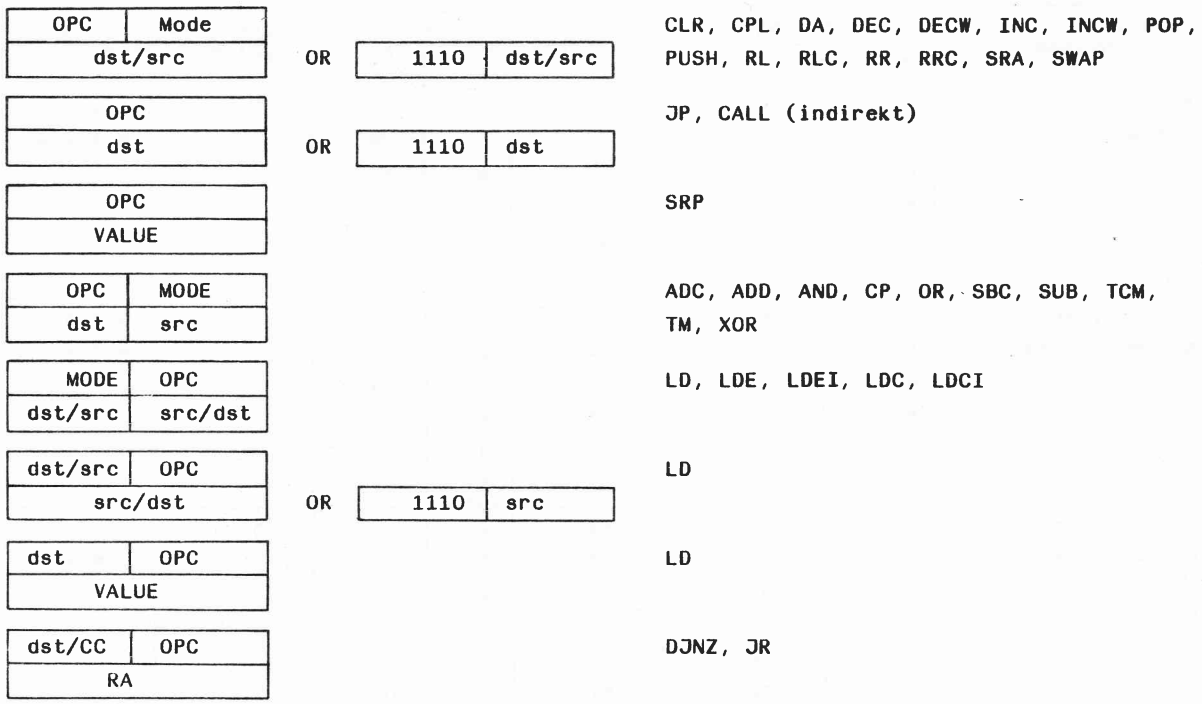


Bild 29: Formate von Zwei-Byte-Befehlen

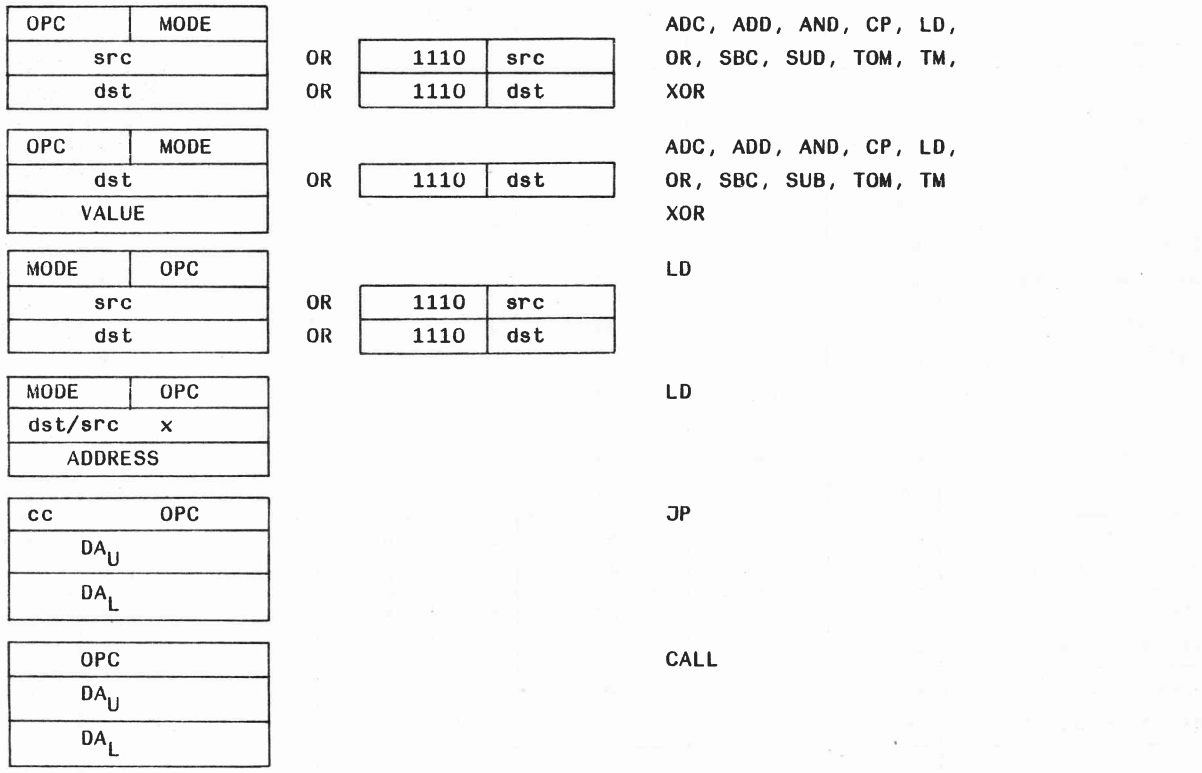
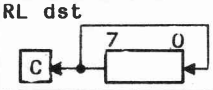
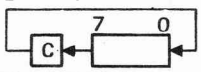
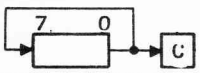
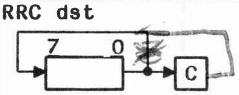


Bild 30: Formate von Drei-Byte-Befehlen

Zusammenfassung der Befehle

Befehl und Funktion	Adressierungsart dst	Adressierungsart src	Hex Operations-code	Ausführung Bytes	Ausführung Zyklen	Pipeline-Zyklen	Flags C	Z	S	V	D	H
ADC dst, src	r	r	12	2	6	5	↕	↕	↕	↕	0	↕
dst ← dst+src+C	r	Ir	13	2	6							
	R	R	14	3	10							
	R	IR	15	3	10							
	R	IM	16	3	10							
	IR	IM	17	3	10							
ADD dst, src	r	r	02	2	6	5	↕	↕	↕	↕	0	↕
dst ← dst+src	r	Ir	03	2	6							
	R	R	04	3	10							
	R	IR	05	3	10							
	R	IM	06	3	10							
	IR	IM	07	3	10							
AND dst, src	r	r	52	2	6	5	•	↕	↕	0	•	•
dst ← dst AND src	r	Ir	53	2	6							
	R	R	54	3	10							
	R	IR	55	3	10							
	R	IM	56	3	10							
	IR	IM	57	3	10							
CALL dst	DA		D6	3	20	0	•	•	•	•	•	•
SP ← SP-2	IRR		D4	2	20	0						
ⓐ SP ← PC												
PC ← dst												
CCF			EF	1	6	5	↕	•	•	•	•	•
C ← $\overline{C}$												
CLR dst	R		B0	2	6	5	•	•	•	•	•	•
dst ← 0	IR		B1	2	6							
COM dst	R		60	2	6	5	•	↕	↕	0	•	•
dst ← $\overline{dst}$	IR		61	2	6							
CP dst, src	r	r	A2	2	6	5	↕	↕	↕	↕	•	•
dst − $\overline{src}$	r	Ir	A3	2	6							
	R	R	A4	3	10							
	R	IR	A5	3	10							
	R	IM	A6	3	10							
	IR	IM	A7	3	10							
DA dst	R		40	2	8	5	↕	↕	↕	X	•	•
dst ← DA dst	IR		41	2	8							
DEC dst	R		00	2	6	5	•	↕	↕	↕	•	•
dst ← dst-1	IR		01	2	6							
DECW dst	RR		80	2	10	5	•	↕	↕	↕	•	•
dst ← dst-1	IR		81	2	10							
DI			8F	1	6	1	•	•	•	•	•	•
IMR (7) ← 0												

Befehl und Funktion	Adressierungsart dst	Adressierungsart src	Hex Operations-code	Ausführung Bytes	Ausführung Zyklen	Pipeline-Zyklen	Flags C Z S V D H
DJNZ r, dst r ← r-1 if r#0 PC ← PC+dst Range: +127, -128	RA		rA r=0-F	2	12/10 (durch-geführt/ nicht durch-geführt)	3/5	• • • • • •
EI IMR (7) ← 1			9F	1	6	1	• • • • • •
INC dst dst ← dst+1	r		rE r=0-F	1	6	5	• ↕ ↕ ↕ • •
	R		20	2	6		
	IR		21	2	6		
INCW dst dst ← dst+1	RR		A0	2	10	5	• ↕ ↕ ↕ • •
	IR		A1	2	10		
IRET Flags ← @ SP SP ← SP+1 PC ← @ SP SP ← SP+2 IMR (7) ← 1			BF	1	16	0	↕ ↕ ↕ ↕ ↕ ↕
JP cc, dst if cc is true, PC ← dst	DA		cD c=0-F	3	12/10 (durch-geführt/ nicht durchge-führt)	0	• • • • • •
	IRR		30	2	8		
JR cc, dst if cc is true, PC ← PC+dst Range: +127, -128	RA		cB c=0-F	2	12/10 (durch-geführt/ nicht durch-geführt)	0	• • • • • •
LD dst, src dst ← src	r	IM	rC	2	6	5	• • • • • •
	r	R	r8	2	6		
	R	r	r9	2	6		
			r=0-F				
	r	x	C7	3	10		
	x	r	D7	3	10		
	r	Ir	E3	2	6		
	Ir	r	F3	2	6		
	R	R	E4	3	10		
	R	IR	E5	3	10		
	R	IM	E6	3	10		
	IR	IM	E7	3	10		
	IR	R	F5	3	10		

Befehl und Funktion	Adressierungsart dst	Adressierungsart src	Hex Operations-code	Ausführung Bytes	Ausführung Zyklen	Pipeline-Zyklen	Flags C Z S V D H
LDC dst src dst ← src	r Irr	Irr r	C2 D2	2 2	12 12	0	• • • • • •
LDCI dst, src dst ← src r ← r+1 rr ← rr+1	Ir Irr	Irr Ir	C3 D3	2 2	18 18	0	• • • • • •
LDE dst, src dst ← src	r Irr	Irr r	82 92	2 2	12 12	0	• • • • • •
LDEI dst, src dst ← src r ← r+1 rr ← rr+1	Ir Irr	Irr Ir	83 93	2 2	18 18		• • • • • •
NOP			FF	1	6	0	• • • • • •
OR dst, src dst ← dst OR src	r r R R R IR	r Ir R IR IM IM	42 43 44 45 46 47	2 2 3 3 3 3	6 6 10 10 10 10	5	• ↕ ↕ 0 • •
POP dst dst ← @ SP SP ← SP+1	R IR		50 51	2 2	10 10	5	• • • • • •
PUSH src SP ← SP-1 @ SP ← src		R IR	70 71	2 2	10/12 (int/ext stack) 12/14 (int/ext stack)	1	• • • • • •
RCF C ← 0			CF	1	6	5	0 • • • • •
RET PC ← @ SP SP ← SP+2			AF	1	14	0	• • • • • •
RL dst 	R IR		90 91	2 2	6 6	5	↕ ↕ ↕ ↕ • •
RLC dst 	R IR		10 11	2 2	6 6	5	↕ ↕ ↕ ↕ • •
RR dst 	R IR		E0 E1	2 2	6 6	5	↕ ↕ ↕ ↕ • •
RRC dst 	R IR		C0 C1	2 2	6 6	5	↕ ↕ ↕ ↕ • •
SBC dst, src dst ← dst-src-C	r r	r Ir	32 33	2 2	6 6	5	↕ ↕ ↕ ↕ 1 ↕

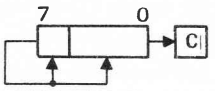
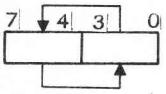
Befehl und Funktion	Adressierungsart dst	src	Hex Operations-code	Ausführung Bytes	Zyklen	Pipeline-Zyklen	Flags C Z S V D H
SBC dst, src dst ← dst-src-C	r	Ir	32	2	6	5	↕ ↕ ↕ ↕ 1 ↕
	r	Ir	33	2	6		
	R	R	34	3	10		
	R	IR	35	3	10		
	R	IM	36	3	10		
	IR	IM	37	3	10		
SCF C ← 1			DF	1	6	5	1 • • • • •
SRA dst 	R		D0	2	6	5	↕ ↕ ↕ 0 • •
	IR		D1	2	6		
SRP src RP ← src		IM	31	2	6	1	• • • • • •
SUB dst, src dst ← dst-src	r	r	22	2	6	5	↕ ↕ ↕ ↕ 1 ↕
	r	Ir	23	2	6		
	R	R	24	3	10		
	R	IR	25	3	10		
	R	IM	26	3	10		
	IR	IM	27	3	10		
SWAP dst 	R		F0	2	8	5	X ↕ ↕ X • •
	IR		F1	2	8		
TCM dst, src (not dst) AND src	r	r	62	2	6	5	• ↕ ↕ 0 • •
	r	Ir	63	2	6		
	R	R	64	3	10		
	R	IR	65	3	10		
	R	IM	66	3	10		
	IR	IM	67	3	10		
TM dst, src dst AND src	r	r	72	2	6	5	• ↕ ↕ 0 • •
	r	Ir	73	2	6		
	R	R	74	3	10		
	R	IR	75	3	10		
	R	IM	76	3	10		
	IR	IM	77	3	10		
XOR dst, src dst ← dst XOR src	r	r	B2	2	6	5	• ↕ ↕ 0 • •
	r	Ir	B3	2	6		
	R	R	B4	3	10		
	R	IR	B5	3	10		
	R	IM	B6	3	10		
	IR	IM	B7	3	10		

Tabelle 5: Zusammenfassung der Befehle

6. Adressierungsarten

Mit Ausnahme der unmittelbaren Daten und Bedingungscodes werden alle Operanden als Registeradressen, Programmspeicheradressen oder Datenspeicheradressen ausgedrückt. Der EMR ermöglicht folgende verschiedene Adressierungsarten:

- Register-Adressierung
- indirekte Register-Adressierung
- indizierte Adressierung
- direkte Adressierung
- relative Adressierung
- unmittelbare Adressierung

6.1. Register-Adressierung

In der Adressierungsart ist der Wert des Operanden der Inhalt des spezifizierten Registers. Das Register kann durch zwei Möglichkeiten adressiert werden:

- durch eine 8-Bit-Adresse im Bereich von 0-127 und 240-255 (Bild 31)
- durch eine 4-Bit-Arbietsregisteradresse im Bereich von 0-15

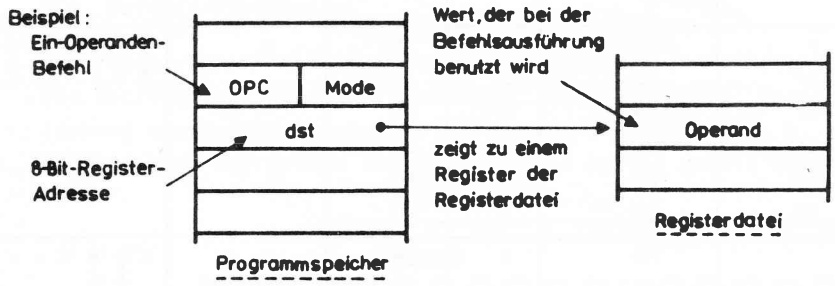


Bild 31: Register-Adressierung

- Arbeitsregister-Adresse:
  Bestimmt man ein Register durch eine 4-Bit-Arbeitsregisteradresse, so verkürzt sich die Länge des Befehls und ergibt eine kürzere Ausführungszeit. In diesem Fall wird die volle (8-Bit-)Registerdatei-Adresse durch Verkettung des 4-Bit-Feldes (Adreßbereich 0-15) mit den oberen 4 Bits des Register-Pointers gebildet. Auf diese Weise kann der Arbeitsregistersatz dynamisch durch Ändern des Wertes des Register-Pointers (R253) variiert werden.

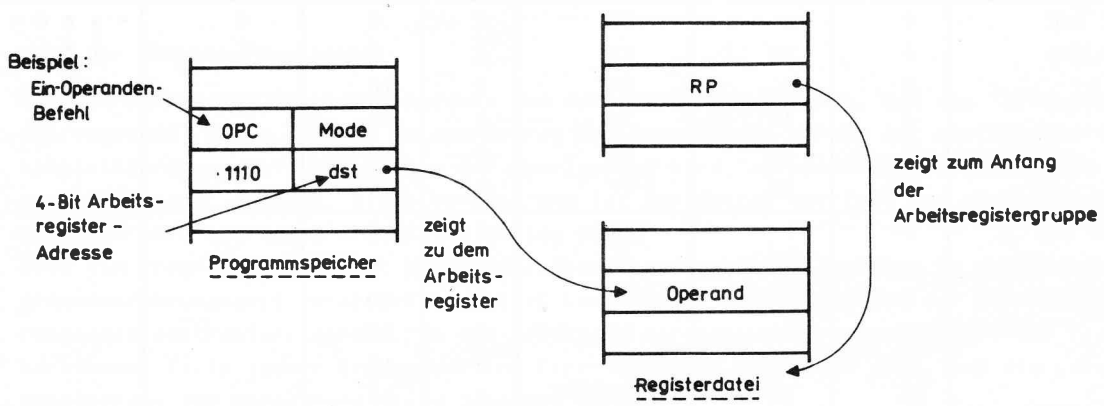


Bild 32: Arbeitsregister-Adressierung

- **Registerpaar-Adresse:**
  Register können auch paarweise benutzt werden, um einen 16-Bit-Wert oder eine Speicheradresse zu kennzeichnen. Ein solches Registerpaar muß geradzahlig im Bereich 0, 2, 4, ..., 126 oder 240, 242, ..., 254 spezifiziert werden.
- **Arbeitsregisterpaar-Adresse:**
  Arbeitsregisterpaare können ebenfalls zur Bestimmung von 16-Bit-Werten oder Speicheradressen verwendet werden. Die zulässigen Registerpaare beginnen geradzahlig und liegen im Bereich 0, 2, 4, ..., 14.

6.2. Indirekte Register-Adressierung

Bei der indirekten Register-Adressierung ist der Wert des Operanden nicht der Inhalt eines Registers. Stattdessen enthalten die Register (Registerpaare, Arbietsregister oder Arbeitsregisterpaare) die Adresse der Speicherzelle, deren Inhalt als Operandenwert verwendet wird. In Abhängigkeit vom ausgewählten Befehl kann die Adresse zu einem Register, zu einer Programmspeicher- oder zu einer externen Datenspeicherzelle zeigen. Registerpaare oder Arbeitsregisterpaare werden zur Speicherung von 16-Bit-Adressen verwendet, wenn man zum Programm- oder zum externen Datenspeicher zugreifen will. Diese Paare werden durch eine gerade Zahl gekennzeichnet.

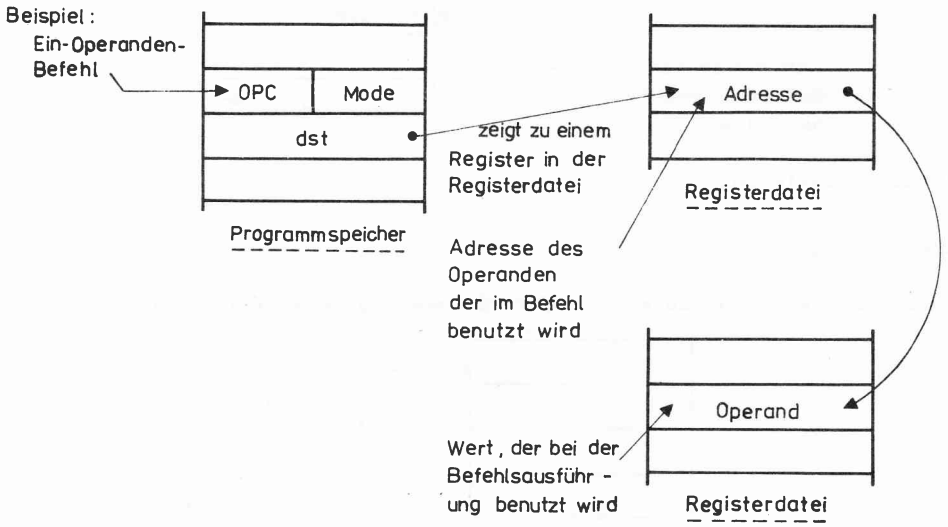


Bild 33: Indirekte Register-Adressierung in der Registerdatei

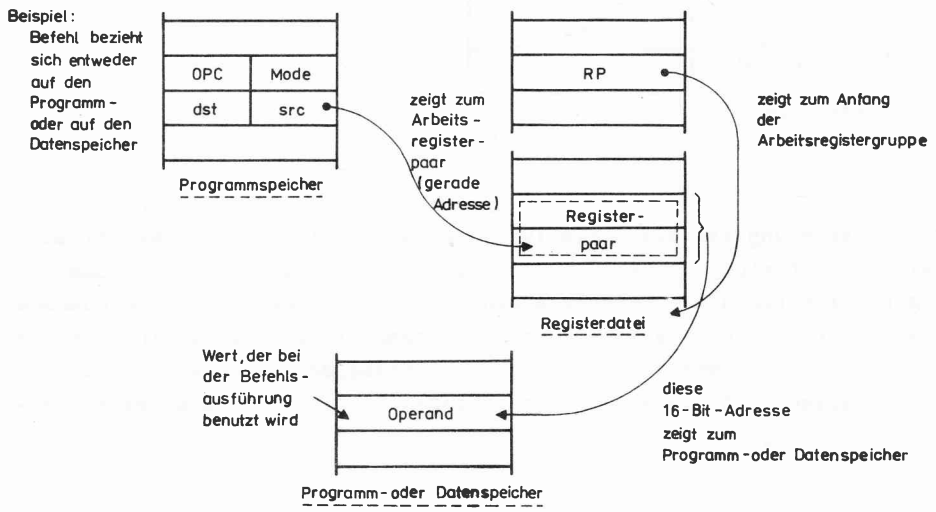


Bild 34: Indirekte Register-Adressierung mit Programm- oder Datenspeicher

6.3. Indizierte Adressierung

Eine indizierte Adresse besteht aus einer Registeradresse, die um den Inhalt eines gekennzeichneten Arbeitsregisters, den Index (Offset), verschoben ist. Dieser Index wird zu der Registeradresse addiert; die resultierende Adresse weist auf die Speicherzelle, deren Inhalt vom entsprechenden Befehl verwendet wird. Diese Adressierungsart wird nur vom Ladebefehl (LD) benutzt, um die Registerdatei zu adressieren.

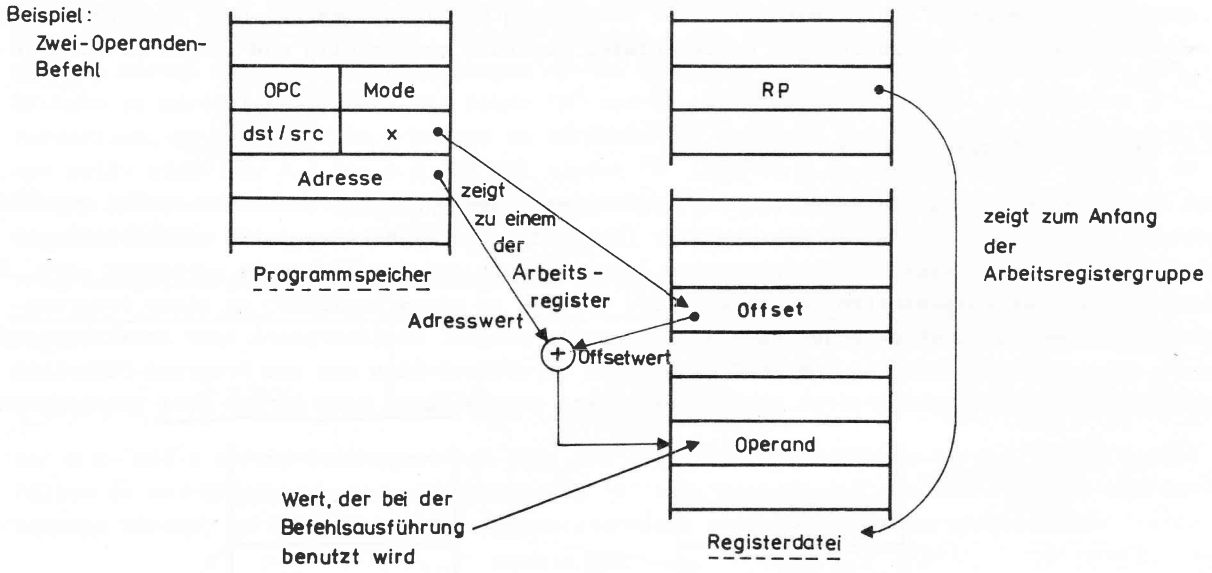


Bild 35: Indizierte Adressierung

6.4. Direkte Adressierung

Die direkte Adressierung wird nur beim bedingten Sprung und beim Call (CALL) benutzt.

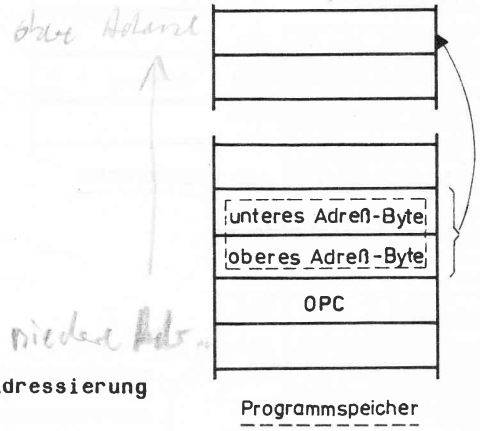


Bild 36: Direkte Adressierung

6.5. Relative Adressierung

Bei der relativen Adressierung ist die (relative) Adresse im eigentlichen Befehl mit einbegriffen. Diese Adressierungsart wird nur bei Jump Relative (JR) und Decrement and Jump (DJNZ) verwendet und ist die einzige für diese Befehle anwendbare Adressierungsart. Der Operand enthält ein Zweier-Komplement-Offset, der zum Inhalt des Programmzählers addiert wird, um die Zieladresse zu bilden. Der Inhalt des Programmzählers ist die Adresse des Befehls, der dem JR- oder DJNZ-Befehl folgt. Der Offsetwert ist ein 8-Bit-vorzeichenbehafteter Wert im Bereich von -128 bis +127.

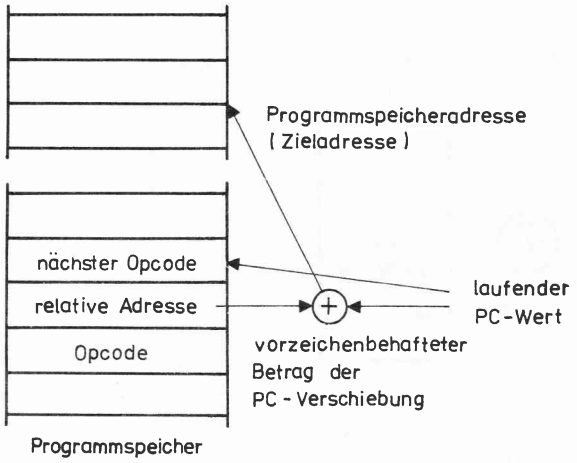


Bild 37: Relative Adressierung

6.6. Unmittelbare Adressierung (Unmittelbare Daten)

Unmittelbare Daten seien hier als eine Adressierungsart betrachtet. Der vom entsprechenden Befehl verwendete Operand wird vom Befehl in seinem Operandenfeld selbst geliefert.

6.7. Anmeldung zum Register-Pointer (RP)

Wenn durch einen Befehl ein vollständiger 8-Bit-"Register-Kennzeichner" gefordert wird (z. B. bei PUSH R), ist ein Verfahren möglich, das erlaubt, die niederen 4 Bits des 8-Bit-"Register-Kennzeichners" in Verbindung mit dem Register-Pointer zu verwenden. So können alle Registerbefehle ein Arbeitsregister oder ein Arbeitsregisterpaar spezifizieren, ausgenommen die Befehle, die entsprechend ihres Formats eine deratige Anwendung ausschließen (siehe Bild 29 und 30). Dieses Verfahren wird durch in folgendem Format spezifizierte Registeradressen angewendet:

1	1	1	0	r_3	r_2	r_1	r_0

Immer wenn das obere Nibbel der Registeradresse ein hexadezimales "E" (1110) enthält, verwendet der EMR automatisch das niedere Nibbel als 4-Bit-Adresse in Verbindung mit dem Register-Pointer, um eine effektive 8-Bit-Adresse zu bilden. Es sind auch Kombinationen von 4-Bit- und 8-Bit-"Kenzeichner" erlaubt (siehe Bild 29 und 30).

7. Applikative Hinweise

7.1. Power-Down-Betrieb

Bei den EMR-Bondvarianten UB 8811 D und UB 8821 M läßt sich unter bestimmten Voraussetzungen die Betriebsspannung U_{CC} abschalten, ohne daß der Inhalt der Mehrzweckregister verlorengeht. An der Stelle des XTAL 2-Ausgangs befindet sich ein Versorgungsspannungseingang (U_{MM}), über den die 124 Mehrzweckregister R4-R127 und die interne Rücksetzlogik versorgt werden. In diesem Fall muß der Takt über den XTAL1-Eingang von einem externen Taktgenerator eingespeist werden.
Zur Sicherung der Daten müssen folgende Punkte beachtet werden:

- Versorgungsspannungsfehler (U_{CC}) müssen extern erkannt werden, um früh genug mittels einer geeigneten Softwareroutine die erforderlichen Daten in die Registerdatei (R4-R127) einzuspeichern!
- $\overline{\text{RESET}}$ muß nach der Datensicherung und solange U_{CC} gestört oder abgeschaltet ist auf "Low" liegen!

• $\overline{\text{RESET}}$ muß zum Schutz der Daten während der Versorgungsspannungszuschaltung auf "Low" liegen!

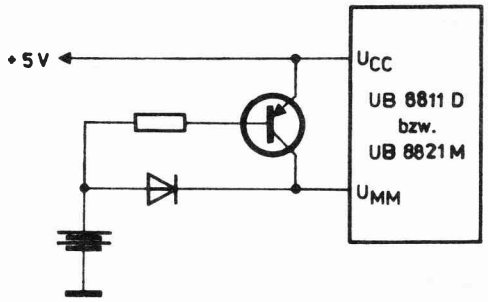


Bild 38: Empfohlene Schaltung für batteriegestütztes System

7.2. Rücksetzen

Zur Initialisierung des EMR-Schaltkreises muß der Rücksetz-Eingang (RESET) für mindestens 50 ms nach dem Anlegen der Versorgungsspannung U_{CC} (Erreichen des U_{CC}-Toleranzbereiches) oder bis 18 Taktzyklen nach Erreichen des U_{CC}-Toleranzbereiches und Stabilisierung des Taktgenerators auf "Low" liegen. Die exakte Zurücksetzung des EMR geschieht durch die in Bild 39 dargestelte externe Beschaltung in Verbindung mit seinem internen "Pull-Up"-Widerstand.

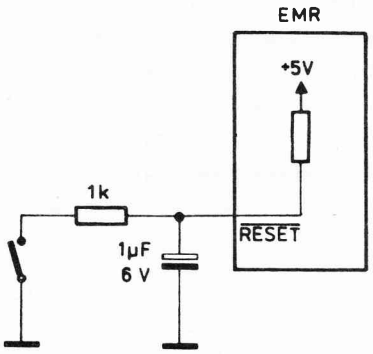


Bild 39: Beschaltung des $\overline{\text{RESET}}$-Einganges bei Versorgungsspannungszuschaltung

Ein Rücksetzen bewirkt, daß die Ports 0, 1, und 2 als Eingänge wirken, ein Setzen des Programmzählers auf 12 und eine Ignorierung von Interruptanforderungen. Die Steuerregister R240-R255 werden gemäß Abschnitt 4.16. initialisiert. Nach einem Rücksetzen beginnt die Programm-Abarbeitung bei der Programmspeicherzelle 12 - beim UB 8810 D/UB 8811 D im internen ROM, beim UB 8820 M/ UB 8821 M im an das Speicherport angeschlossenen Programmspeicher. $\overline{\text{RESET}}$ wird auch bei Power-Down-Betrieb (Abschnitt 7.1.) und beim Testbetrieb (Abschn. 7.4.) benutzt.

7.3. Takt

Der Oszillator auf dem EMR-Chip kann durch einen Quarz mit einem Serienresonanzwiderstand kleiner als 100 Ohm (Schaltungsvorschlag Bild 40) oder durch eine externe Taktquelle (Schaltungsvorschlag Bild 41) angesteuert werden. Dieser On-Chip-Oszillator besitzt einen Serienresonanzverstärker mit hoher Verstärkung. XTAL 1 stellt den Eingang und XTAL 2 den Ausgang des Oszillators dar.

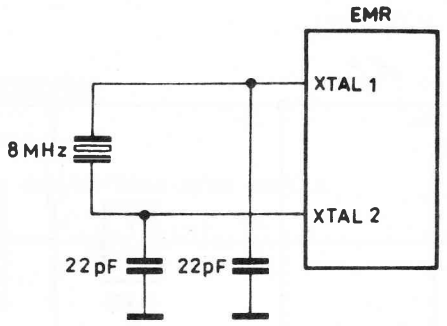


Bild 40: Schaltungsvorschlag für Beschaltung mit Quarz

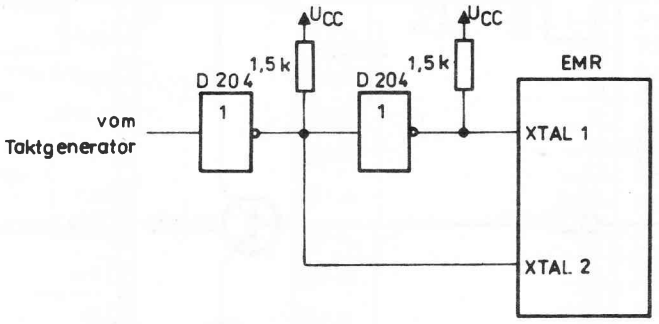


Bild 41: Schaltungsvorschlag für Beschaltung mit externer Taktquelle

Der eigentliche Systemtakt ergibt sich intern aus der durch 2 geteilten Quarz- bzw. Taktgenerator-Frequenz.

7.4. Testbetrieb beim U 8810 D/UB 8811 D

Die EMR-Schaltkreise mit internem ROM UB 8810 D und UB 8811 D enthalten neben dem eigentlichen 2 KByte-Programmspeicher einen 64 Byte-Testprogrammspeicher. Dieses Testprogramm ist erforderlich, um ohne Nutzung (und ohne Kenntnis) des 2 KByte-Programmspeichers die Funktion des EMR testen zu können. Um in diesen Testbetrieb zu gelangen, um den Testprogrammspeicher ansprechen zu können, muß der $\overline{\text{RESET}}$-Eingang nach Beendigung des Rücksetz-Vorgangs (siehe Abschnitt 7.2.) sofort auf einen Pegel oberhalb von +7 Volt angehoben werden. Diese Maßnahme gibt den Test-ROM-Speicherzellen 0 bis 63 den Vorrang gegenüber dem Anwenderprogrammspeicher. Die ROM-Speicherzellen oberhalb 63 (d. h. 64-2047) werden durch den normalen Anwender-ROM verkörpert. Der Testbetrieb läß sich unter bestimmten Bedingungen vom Anwender ausnutzen, um mittels eines externen Programmspeichers (über Port 0 und 1) einen EMR ohne Nutzung des intern angeordneten ROM's betreiben zu können. Bild 42 zeigt eine Möglichkeit der Beschaltung von XTAL1, XTAL2 und RESET zum Erreichen des Testbetriebes.
Nach dem Abarbeiten des Test-ROM's weist der Programmzähler auf die Adresse 812_H, die im externen Speicherbereich liegt. An dieser Stelle kann vom Anwender der erste Befehl seines Programms angeordnet werden. Der Testbetrieb kann jederzeit durch ein normales Rücksetzen verlassen werden. Der Inhalt des Test-ROM's wird ausschließlich durch den Hersteller festgelegt und ist kein Bestandteil garantierter Schaltkreisfunktionen (Typstandards). Eine Ausnahme bilden die vom veb mikroelektronik "karl marx" erfurt angebotenen Typen UB 8860 D und UB 8861 D, bei denen die Funktion des Test-ROM's garantiert wird.

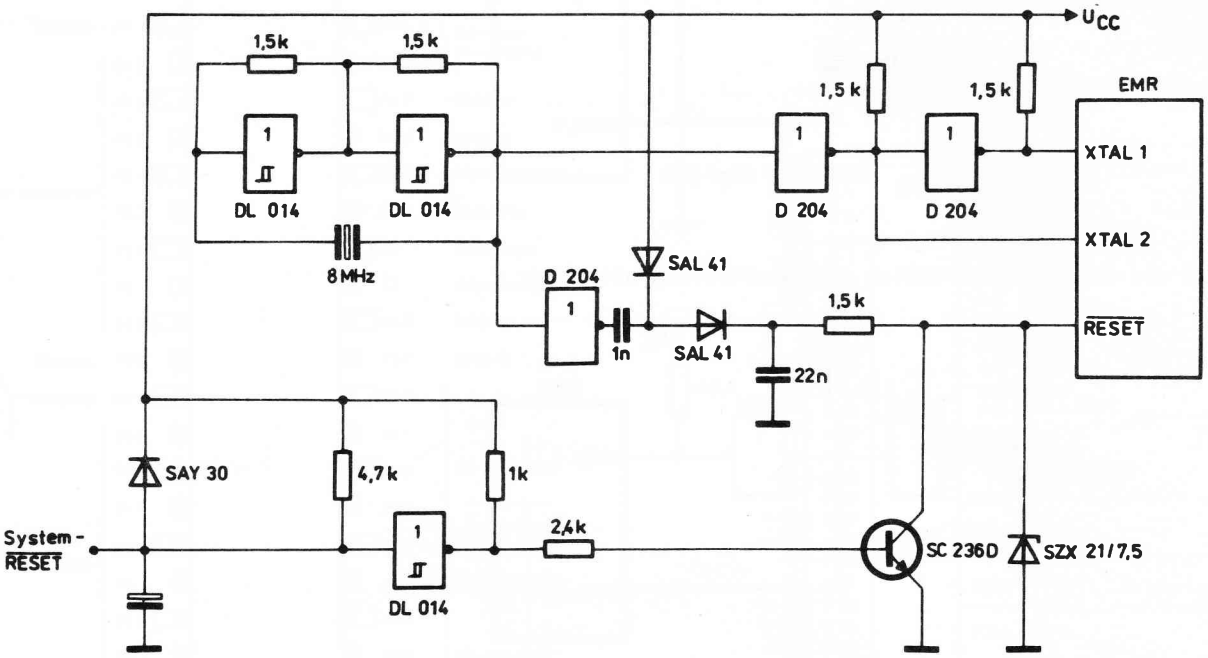


Bild 42: Schaltungsvorschlag für den Eintritt in den Testbetrieb beim UB 8810 D/ UB 8811 D

8. Elektrische Parameter

Alle im folgenden Abschnitt aufgeführten elektrischen Parameter beziehen sich - soweit nicht anders angegeben - auf die Typen UB 8810 D/UB 8811 D und UB 8820 M/UB 8821 M.

Angaben zu den entsprechenden Einstellwerten, Prüfkategorien, Bewertungskriterien und Meßschaltungen sind den jeweiligen Typstandards (TGL) zu entnehmen.

8.1. Haupt- und Nebenkenngrößen (Auswahl)

Kenngröße	Kurzzeichen	Einheit	Kleinstwert	Größtwert
Eingangsreststrom	I_{IL}	µA	-	/20/
Eingangsreststrom RESET	$-I_{ILR}$	µA	-	80
Ausgangsreststrom	I_{ILO}	µA	-	/20/
L-Ausgangsspannung *1) bei Belastung	U_{OL}	V	-	0,4
H-Ausgangsspannung *2) bei Belastung	U_{OH}	V	2,4	-
Stromaufnahme	I_{CC}	mA	-	200
statische Stromauf- *3) nahme	I_{MM}	mA	-	20

Anmerkung

*1) Für A0 bis A11, $\overline{MDS}$, $\overline{SYNC}$, SCLK und IACK beim UB 8820 M/UB 8821 M gilt U_{OL} für I_{OL} = 1 mA, in allen anderen Fällen für I_{OL} = 2 mA.

*2) Für A0 bis A11, $\overline{MDS}$, $\overline{SYNC}$, SCLK und IACK beim UB 8820 M/UB 8821 M gilt U_{OH} für I_{OH}= -0,1 mA, in allen anderen Fällen für I_{OH}= -0,25 mA.

*3) Gilt für UB 8811 D und UB 8821 M bei U_{MM}= 3 V.

Kenngröße *4)	Kurzzeichen	Einheit	Kleinstwert	Größtwert	Anmerk.
Adressen gültig bis Adressenstrobe	$t_{dA(AS)}$	ns	50	-	*5)
Adressenstrobe bis Adressen ungültig	$t_{dAS(A)}$		70	-	
Adressenstrobe bis Eingabedaten gültig	$t_{dAS(DI)}$		-	360	*6)
Adressenstrobe-Länge	t_{wAS}		80	-	*5)
Adressen ungültig bis Datenstrobe	$t_{dA(DS)}$		0	-	-
Datenstrobe-Länge Read	t_{wDS}		250	-	*5)
Datenstrobe-Länge Write	t_{wDS}		160	-	*7)
Datenstrobe bis Eingabedaten gültig	$t_{dDS(DI)}$		-	200	*6)
Eingabedaten Haltezeit	$t_{hDS(DI)}$		0	-	-
Datenstrobe bis Änderung der Adressen	$t_{dDS(A)}$		80	-	*5)
Datenstrobe bis Adressenstrobe	$t_{dDS(AS)}$		70	-	
Read gültig bis Adressenstrobe	$t_{dR(AS)}$		50	-	
Datenstrobe bis Änderung der Eingabedaten	$t_{dDS(R)}$		60	-	
Ausgabedaten gültig bis Datenstrobe	$t_{dDO(DS)}$		50	-	
Datenstrobe bis Änderung der Ausgabedaten	$t_{dDS(DO)}$		80	-	
Write gültig bis Adressenstrobe	$t_{dw(AS)}$		50	-	
Datenstrobe bis Änderung von Write	$t_{dDS(W)}$		60	-	

Kenngröße *4)	Kurzzeichen	Einheit	Kleinstwert	Größtwert	Anmerk.
Eingangsdaten Setzzeit	$t_{sDI(DA)}$	ns	0	-	
Eingangsdaten Haltezeit	$t_{hDA(DI)}$		230	-	
Länge von $\overline{DAV}$ Eingabe	t_{wDA}		175	-	
DAV=L bis $\overline{RDY}$ Eingabe	$t_{dAL(RY)}$		20	175	
$\overline{DAV}$=L bis $\overline{RDY}$ Ausgabe	$t_{dAL(RY)}$		0	-	*8)
$\overline{DAV}$=H bis $\overline{RDY}$ Eingabe	$t_{dDAH(RY)}$		0	150	*9)
$\overline{DAV}$=H bis $\overline{RDY}$ Ausgabe	$t_{dDAH(RY)}$		0	-	
Datenausgabe bis $\overline{DAV}$	$t_{dDO(DA)}$		50	-	
RDY bis $\overline{DAV}$	$t_{dRY(DA)}$		0	205	*8)
Impulslänge für externe Interruptanforderung	$t_{wI(L)}$		100	-	
Pause zwischen externen Interruptanforderungen	$t_{wI(H)}$		250	-	
Systemtakt-Ausgabe bis Adressenstrobe	$t_{dSC(AS)}$		-	0	
SYNC-Ausgabe bis Datenstrobe	$t_{dSY(DS)}$		200	-	
Länge der SYNC-Ausgabe	t_{wSY}		160	-	*10)
Adressen gültig bis Eingangsdaten (Speicher-Port)	$t_{dA(DI)}$		-	450	
Eingangsdaten-Haltezeit (Speicher-Port)	$t_{hDI(A)}$		0	-	

Anmerkungen:

*4) Alle Zeitbeziehungen gelten für 2 V für logisch "H" und 0,8 V

*5) Die Zeiten sind für eine Eingangstaktfrequenz von 8 MHz spezifiziert. Wenn mit einer geringeren Taktfrequenz gearbeitet wird, ist der Zuwachs in einer Taktperiode zu addieren.

*6) Diese Verzögerungszeiten stellen Zugriffszeiten zum Systemspeicher dar und gelten für 8 MHz Eingangsfrequenz, Für geringere Frequenzen muß die Änderung während 4 Taktperioden zu $t_{dAS(DI)}$ und während 3 Taktperioden zu $t_{dDS(DI)}$ addiert werden.

*7) Die Länge von Datenstrobe ist abhängig von der Länge des ausgeführten Befehls.

*8) Jede dieser beiden Zeiten kann einzeln betrachtet 0 ns betragen:
$t_{dDAL(RY)} + t_{dRY(DA)} \geq 2\ t_{pC}$ (für 8 MHz mindestens 250 ns)

*9) Es müssen zwei Bedingungen erfüllt sein, bevor RDY von "L" auf "H" geht:
1. Der $\overline{DAV}$-Eingang muß von "L" auf "H" gegangen sein.
2. Die Eingabedaten müssen vom Port-Register in den EMR übernommen werden (z. B. LD 4,2 : Die Eingabedaten werden vom Port 2-Register in Register 4 übernommen).

Spätestens 150 ns nach Erfüllung beider Bedingungen geht der RDY-Ausgang von "L" auf "H".

*10) Gilt nur für UB 8820 M/UB 8821 M.

8.2. Grenzwerte (Auswahl)

Kenngröße	Kurzzeichen	Einheit	Kleinstwert	Größtwert
Betriebsspannung	U_{CC}	V	-0,5	7
Eingangsspannung	U_I	V	-0,5	7
Ausgangsspannung	U_O	V	-0,5	7

Alle Werte beziehen sich auf U_{SS}= 0 V.

8.3. Statische Betriebsbedingungen

Kenngröße	Kurzzeichen	Einheit	Kleinstwert	Größtwert
Betriebsspannung (Arbeitsbetrieb)	U_{CC}	V	4,75	5,25
	U_{MM}	V	U_{CC}-0,6	U_{CC}
Betriebsspannung (Power-Down-Betrieb)	U_{CC}	V	0	4,749
	U_{MM}	V	3	5,25
Eingangsspannung	U_{IL}	V	-0,3	0,8
	U_{IH}	V	2	U_{CC}
Takteingangsspannung *11)	U_{ILC}	V	-0,3	0,8
	U_{IHC}	V	3,8	U_{CC}
RESET-Eingangsspannung	U_{ILR}	V	-0,3	0,8
	U_{IHR}	V	3,8	U_{CC}
Betriebstemperaturbereich	ϑ_a	°C	0 bis 70	

Anmerkung:

*11) vom externen Taktgenerator

8.4. Dynamische Betriebsbedingungen

Kenngröße	Kurzzeichen	Einheit	Kleinstwert	Größtwert
Eingangstaktfrequenz	f_C	MHz	1	8
Eingangstakt-Anstiegs-und Abfallzeiten *11)	t_{rC}, t_{fC}	ns	-	25
Taktbreite *11)	t_{WC}	ns	37	-

Anmerkung:

*11) vom externen Taktgenerator

8.5. Bilder zum Zeitverhalten

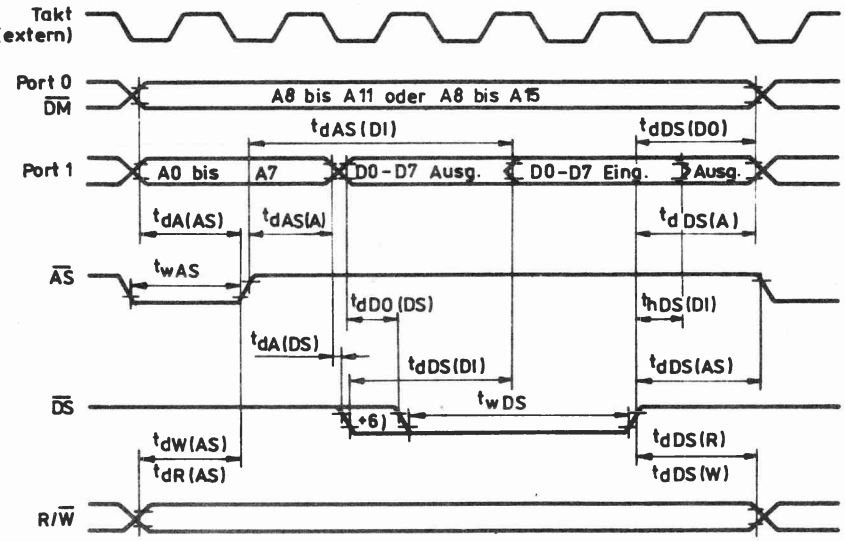


Bild 43 Zeitverhalten für externe Ein-/Ausgabe oder Speicherlesen und -schreiben

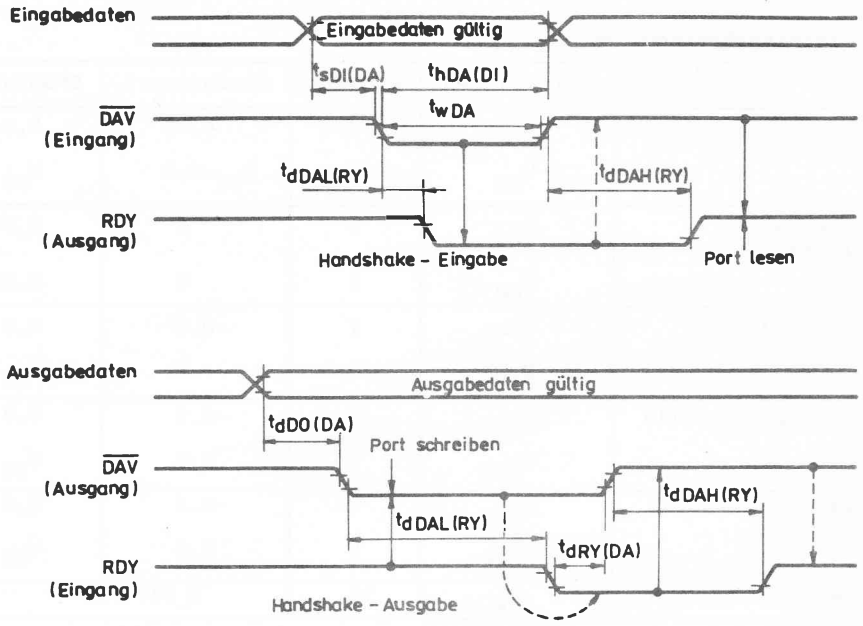


Bild 44 Handshake-Zeitverhalten

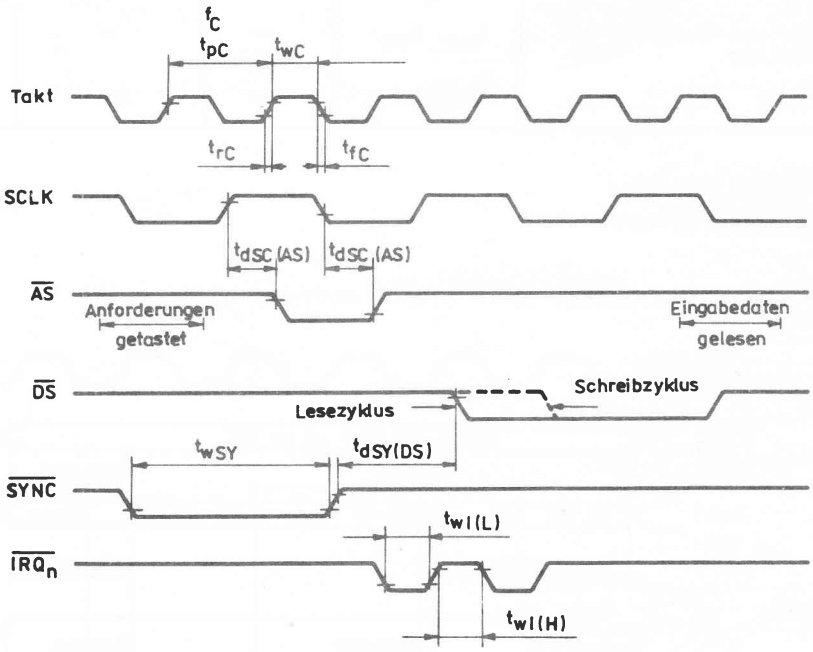


Bild 45 Allgemeines Zeitverhalten

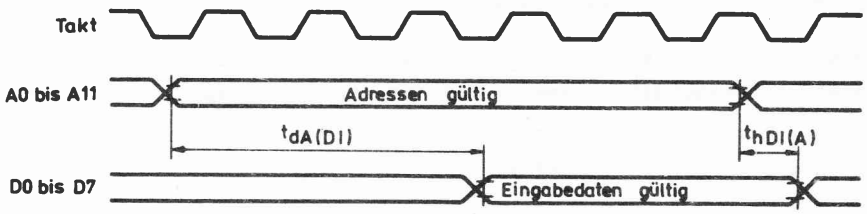


Bild 46 Speicher-Port-Zeitverhalten
(gilt nur für UB 8820 M/UB 8821 M)

9. Typstandards

Verbindliche technische Liefer- und Reklamationsgrundlage für oben beschriebene EMR-Schaltkreise bilden ausschließlich die entsprechenden Typstandards (TGL):

UB 8810 D, UB 8811 D	TGL 37360
UB 8820 M, UB 8821 M	TGL 42639

Die nachstehende Übersicht umfaßt das gesamte vom veb mikroelektronik "karl marx" erfurt angebotene Sortiment an EMR-Schaltkreisen, unter Hinweis auf entsprechende Typstandards:

UB 8810 D, UB 8811 D	TGL 37360
UD 8810 D, UD 8811 D	TGL 42641
UB 8820 M, UB 8821 M	TGL 42639
UC 8820 M, UC 8821 M	
UD 8820 M, UD 8821 M	TGL 42640
UB 8830 D, UB 8831 D	TGL 38607
UC 8830 D, UC 8831 D	TGL 38609
UD 8830 D, UD 8831 D	TGL 38608
UB 8840 M, UB 8841 M	
UC 8840 M, UC 8841 M	TGL 42634
UD 8840 M, UD 8841 M	
UB 8860 D, UB 8861 D	
UC 8860 D, UC 8861 D	TGL 37359
UD 8860 D, UD 8861 D	

RS 1286/86 V7 1658 N2

RFT

veb mikroelektronik ›karl marx‹ erfurt
stammbetrieb

DDR – 5010 Erfurt, Rudolfstraße 47 Telefon 580 Telex 061306

elektronik
export·import
Volkseigener Außenhandelsbetrieb der
Deutschen Demokratischen Republik
DDR - 1026 Berlin, Alexanderplatz 6
Telex: BLN 114721 elei, Telefon: 2180