



SONDER INFO

IHD 164180

Interne Ports

MMU

DMAC

Serielle Schnittstellen

Timer

Die internen Ports des HD64180

Helmut Bernhardt

01

Sämtliche gegenüber dem Z80 zusätzlich im HD64180 zur Verfügung gestellten Leistungen sind über Portadressen zu programmieren. Für diese internen Ports der CPU wird ein I/O-Adreßbereich von 3FH Ports verbraucht.

Wie prinzipiell auch schon beim Z80 kann mit dem HD64180 ein I/O-Adreßraum von 64K-Ports differenziert werden. Beim Z80 werden dafür ausschließlich die Befehle IN A,(C) und OUT (C),A sinnvoll sein. Bei diesen Befehlen werden die Inhalte der Register C (A0-A7) und B (A8-A15) als Adressen auf den Bus gelegt. Bei den normalen IN- und OUT-Befehlen mit Angabe einer 8Bit-Portadresse im Befehlsoperanden wird der Inhalt des Accu auf A8-A15 des Adreßbus gelegt. Diese Befehle sind in Systemen, deren I/O-Adreßdecoding auf 16Bit-Adressen aufbaut, nicht verwendbar.

Die internen Ports des HD64180 liegen in der Zero-Page (A8-A15 sind low) des 64K-I/O-Adreßraums. Um diese internen Ports zu lesen und zu schreiben, hat der HD64180 spezielle Befehle, die automatisch A8-A15 low ausgeben, wenn auf die interne Ports zugegriffen wird.

Traditionsbedingt decodieren aber sämtliche I/O-mapped Peripherie-Geräte für Z80-Systeme nur A0-A7 und beschränken den I/O-Adreßraum damit auf 256 Ports. Diese Baugruppen fühlen sich bei Zugriffen auf die internen Ports ebenfalls angesprochen, wenn ihre 8Bit-Adresse mit der Zero-Page-Adresse des angesprochenen internen Ports übereinstimmt. Beim Lesen eines internen Ports werden deshalb die Datentreiber des HD64180-Chips zur Umwelt geschlossen, so daß keine Kollisionen von internen und externen Daten möglich sind. Beim Schreiben gelangen aber die Daten, die für die internen Ports bestimmt sind, auch nach außen und werden von einem auf gleicher Adresse liegenden externen Port auch vereinnahmt. Verrückt spielende Peripherie-Geräte wären die Folge, wenn nicht noch eine zusätzliche Sicherung im HD64180 eingebaut wäre.

Der zusammenhängende Adreßbereich von 3FH Zero-Page-Ports für die Register des HD64180 beginnt default bei 00H. Durch ändern der Bits 7 und 6 des ICR-Registers (3FH) lassen sich die 3FH Ports innerhalb der Zero-Page verschieben.

D7	D6	Adreßbereich der internen Ports
0	0	00 - 3F (nach RESET)
0	1	40 - 7F
1	0	80 - BF
1	1	C0 - FF

Die im folgenden benutzte Angabe von Portadressen geht davon aus, daß D6 und D7 von ICR beide low sind und die internen Ports bei 00-3F in der Zero-Page liegen. Die relativen Adressen und Funktionen sind in der Tabelle aus dem Prof180-Handbuch zu ersehen. Mit dieser Tabelle alleine ist man aber noch lange nicht in der Lage, die internen Ports sinnvoll zu programmieren. Vielleicht helfen die folgenden Ausführungen da weiter:

Zunächst seien die Angaben zum ICR-Register (I/O Control Register,

Port 3FH) vervollständigt. D5 (IOSTP) = 0 ermöglicht den normalen Betrieb der HD64180-internen Peripherie-Ports (Einstellung nach RESET). Wenn IOSTP auf 1 gesetzt wird, sind die Funktionen von ASCII, CSI/O und PRT (serielle Schnittstellen und Reload-Timer) unterbunden.

02

Damit ein bei 9,216MHz Taktrate tickender HD64180 auch mit langsamer Peripherie laufen kann, gibt es nicht nur den WAIT-Pin, über den die Peripherie die CPU um Geduld bitten kann; die CPU (und auch die internen DMA-Kanäle) können auch selbständig WAIT-Zyklen einlegen, wobei zwischen memory-mapped und I/O-mapped Zugriffen unterschieden wird. Mit dem oberen Nibble von Port 32H (DCNTL) lassen sich unterschiedliche Anzahlen WAITs einstellen:

D7	D6	Anzahl WAITs bei memory mapped Adressierung
0	0	0
0	1	1
1	0	2
1	1	3 (nach RESET)

D5	D4	Anzahl WAITs bei I/O-Zugriffen /INTO-Acknowledge	
0	0	1	2
0	1	2	4
1	0	3	5
1	1	4	6 (nach RESET)

Bei Zugriff auf die internen Ports werden unabhängig von D5 und D4 keine WAITs eingefügt. Nur für die Datenregister von ASCII, CSI/O und PRT werden je nach Zustand bis zu 4 WAITs für die Synchronisation mit der CPU eingefügt.

Die Bits 0-3 von DCNTL dienen der Steuerung des DMA-Betriebs und werden dort behandelt.

RCR, Refresh Control Register, Port 36H

Gegenüber dem Hidden Refresh des Z80 während jeden M1-Zyklus' gibt der HD64180 jeweils nach einer bestimmten Anzahl von Taktzyklen eine 8Bit-Refreshadresse aus. Durch die 8Bit-Adresse können 256K-Bit- und 1M-Bit-Speicher-Chips ohne zusätzliche Refresh-Logik verwendet werden. Ein Refresh erfolgt asynchron zur CPU-Arbeit nach einer über RCR, D0 und D1 eingestellten Anzahl von Taktzyklen:

D1	D0	Anzahl Takte zwischen zwei Refresh-Zyklen
0	0	10 (nach RESET)
0	1	20
1	0	40
1	1	80

Mit D7=0 an RCR wird der Refresh-Controller abgeschaltet (für statische RAMs kann das Refreshing entfallen. D7=1 (nach RESET) aktiviert den Refresh-Controller.

Mit D6 läßt sich das Refresh Timing den Möglichkeiten des Speichers anpassen:

- D6 = 0 : Refresh ist 2 Taktzyklen lang
- D6 = 1 : Refresh ist 3 Taktzyklen lang (nach RESET voreingestellt)

D2-D5 im RCR-Register haben keine Funktion.

ICT, Interrupt/Trap Register, Port 34H

D7 (Trap) wird immer dann von der CPU auf 1 gesetzt, wenn diese auf einen illegalen Befehl gestoßen ist (z.B. undefinierte Z80-Befehle). Durch einen OUT0-Befehl kann D7 wieder auf 0 gesetzt werden (nicht aber auf 1). Nach RESET ist D7=0.

Ein illegaler OP-Code löst den für den HD64180 höchst priorisierten Trap-Interrupt aus. Die Service-Routine kann dann aus dem Zustand von D6 (UFO) erkennen, ob der Fehler im 2ten oder 3ten Byte des OP-Codes auftrat.

D6 erlaubt die Korrektur des beim Trap auf den Stack geretteten Program-Counters. Bei D6=0 zeigt der gerettete PC auf das zweite Byte des als falsch erkannten Befehls, bei D6=1 zeigt der PC auf das 3. Byte des falschen Befehls. D6 kann nur gelesen werden.

D0, D1 und D2 können zur Maskierung der externen Interrupts /INT0, /INT1 und /INT2 verwendet werden. Wenn das entsprechende Bit auf 0 gesetzt ist, ist der Interrupt gesperrt; auf 1 gesetzt wird der entsprechende Interrupt freigegeben. Nach RESET ist nur /INT0 freigegeben (D0=1) und /INT1 und /INT2 sind gesperrt (D1,D2=0).

IL, Interrupt Vector Low Register, Port 33H

Die externen Interrupts /INT0 im Mode 2 sowie /INT1 und /INT2 grundsätzlich und alle internen Interrupts werden von Service-Routinen bedient, deren Startadressen in einer Tabelle abgelegt sind. Die Technik ist die des IM2 beim Z80. Im I-Register der CPU werden die oberen 8 Bit (A8-A15) der Adresse der Vektor-Tabelle abgelegt. Die unteren 8 Bit der Adresse (Pointer in die Vektor-Tabelle) liefert das Interrupt-auslösende Peripherie-Gerät beim Interrupt-Acknowledge über den Datenbus. Die CPU ignoriert davon D0, um eine gerade Adresse zu erzeugen, und bildet aus den beiden Hälften einen Pointer in die Vektor-Tabelle. Die dort in der Tabelle abgelegte Adresse ist die Startadresse der Service-Routine.

Beim HD64180 benötigt nur der /INT0, der dem /INT des Z80 im Mode 2 entspricht, einen extern gelieferten Adreßanteil (A1-A7 über D1-D7) des Datenbus), um in der Vektor-Tabelle die Adresse der Service-Routine zu finden. Damit können abwechselnd mehrere INT-Quellen über nur einen Interrupt-Eingang der CPU angeschlossen werden.

Die Adressen der übrigen Interrupt-Quellen des HD64180 sind (mit gewissen Freiheiten) festgelegt.

So stellen die ersten beiden Bytes der Tabelle (Adresse mit A0-A7 = 0) die Adresse mit der Service-Routine für /INT1 dar. Die nächsten beiden

Bytes liefern den Vektor für /INT2. Die Reihenfolge der Vektoren ist durch die Interrupt-Quellen festgelegt; es läßt sich aber die relative Lage der Gruppe von Vektoren (32 Bytes) innerhalb der 256-Bytes-Tabelle in 32-Bytes-Schritten verschieben. Wo diese Adressengruppe relativ in der Tabelle liegt, wird durch D7-D5 des IL-Registers vorgegeben (nach RESET sind diese low und die Gruppe liegt am Anfang der Vektor-Tabelle).

Wenn D7-D5 alle low sind, liegen die Vektoren für /INT1, /INT2 und die internen Interrupts ab Adresse XX00H in der Tabelle; wenn D7-D5 das Bitmuster 001 haben, liegen diese Vektoren ab XX20H in der Tabelle; u.s.w.

Die Bits 0-4 werden von der internen Interrupt-Quelle selbst erzeugt und geben die relative Adresse innerhalb der 32Byte-Gruppe in der Tabelle vor. Die Tabelle 2.7.2 des HD64180-Handbuchs (die hier ohne freundliche Genehmigung von Hitachi wiedergegeben ist) gibt die Reihenfolge der Einträge für die Interrupt-Quellen und die damit verknüpfte Prioritäts-Hierarchie vor.

Table 2.7.2 Interrupt Source and Lower Vector

Interrupt-Source	Priority	IL			Fixed Code				
		b7	b6	b5	b4	b3	b2	b1	b0
/INT1	highest	.	.	.	0	0	0	0	0
/INT2		.	.	.	0	0	0	1	0
PRT channel 0		.	.	.	0	0	1	0	0
PRT channel 1		.	.	.	0	0	1	1	0
DMA channel 0		.	.	.	0	1	0	0	0
DMA channel 1		.	.	.	0	1	0	1	0
CSI/0		.	.	.	0	1	1	0	0
ASCI channel 0		.	.	.	0	1	1	1	0
ASCI channel 1 lowest		.	.	.	1	0	0	0	0

Außer den maskierten Interrupts gibt es noch den oben erwähnten Trap, der eine Verzweigung zur logischen Adresse 0000H zur Folge hat. Wenn diese logische Adresse mit der physikalischen Adresse 00000H identisch ist (Banking), entspricht dies dem /RESET-Vektor. Durch Prüfen von D7 in Port 32H (ICT) kann erkannt werden, ob ein Trap oder ein RESET vorliegt.

Ein /NMI bewirkt wie beim Z80 einen CALL 66H, wobei auch hier die logische Adresse innerhalb der eingestellten Bank und nicht unbedingt die absolute Adresse 00066H relevant ist.

Der /INT0 kennt wie der /INT des Z80 noch die Modi 0 und 1.

Mode 0 : Beim Acknowledge wird ein 1Byte-Befehl vom Datenbus gelesen und ausgeführt (RST XX). Dieses Byte muß die Interrupt-Quelle liefern.

Mode 1 : Bei einem Interrupt am /INT0-Pin erfolgt ein RST 38H. Die Adresse 0038H liegt in der per MMU eingestellten Bank (logische Adresse).

Weitere Steuerungen der internen Interrupts werden bei der Beschreibung der jeweiligen Funktionsgruppen gezeigt. Dazu aber erst im nächsten Info.

Table 2.5.1 Internal I/O Register Address Map (1)

05

	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCI	ASCI Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCI Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCI Control Register B Ch 0	CNTLB0	XX000010	02H
	ASCI Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCI Status Register Ch 0	STAT0	XX000100	04H
	ASCI Status Register Ch 1	STAT1	XX000101	05H
	ASCI Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCI Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCI Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCI Receive Data Register Ch 1	RDR1	XX001001	09H
CS/I/O	CS/I/O Control Register	CNTR	XX001010	0AH
	CS/I/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch 0L	TMDROL	XX001100	0CH
	Timer Data Register Ch 0H	TMDROH	XX001101	0DH
	Reload Register Ch 0L	RLDR0L	XX001110	0EH
	Reload Register Ch 0H	RLDR0H	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001	11H
			XX010011	13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
Reload Register Ch 1L	RLDR1L	XX010110	16H	
Reload Register Ch 1H	RLDR1H	XX010111	17H	
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			XX011111	1FH
DMA	DMA Source Address Register Ch 0L	SAR0L	XX100000	20H
	DMA Source Address Register Ch 0H	SAR0H	XX100001	21H
	DMA Source Address Register Ch 0B	SAR0B	XX100010	22H
	DMA Destination Address Register Ch 0L	DAR0L	XX100011	23H
	DMA Destination Address Register Ch 0H	DAR0H	XX100100	24H
	DMA Destination Address Register Ch 0B	DAR0B	XX100101	25H
	DMA Byte Count Register Ch 0L	BCR0L	XX100110	26H
	DMA Byte Count Register Ch 0H	BCR0H	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Reserved		XX101101	2DH
	DMA Byte Count Register Ch 1L	BCR1L	XX101110	2EH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
DMA/WAIT Control Register	DCNTL	XX110010	32H	
INT	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H
	INT/TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H

06

	Register	Mnemonic	Address	
			Binary	Hexadecimal
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
MMU	MMU Common Base Register	CBR	XX111000	38H
	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
I/O	Reserved		XX111011	3BH
			XX111110	3EH
	I/O Control Register	ICR	XX111111	3FH



Die seriellen Schnittstellen (ASCI und CSI/O) des HD64180
von Gerald Schröder

Der HD64180 stellt drei serielle Schnittstellen zur Verfügung. Bei zweien handelt es sich um "Asynchronous Serial Communication Interfaces" (wie RS-232/V24) und die dritte heißt "Clocked Serial Input/Output".

Asynchronous Serial Communication Interfaces

Aufgabe dieser beiden Schnittstellen ist die serielle Verbindung zu einem Modem, Terminal, (RS-232-)Drucker o.ä. Zusätzlich gibt es einen "Multiprocessor-Modus", mit dem sich leicht ein System mit mehreren HD64180s aufbauen ließe, wobei die ASCIs dann als (Verbindung zum) System-Bus dienen würden.

Jeder Kanal belegt 5 Ports: einen zum Lesen und einen zum Schreiben der Daten, einen für den Status (Zustand) des Kanals, zwei für die Einstellung des Kanals (Baudrate usw.) und einige Zusatz-Informationen. Die Kanäle unterscheiden sich in einigen Features, was ich aber jeweils anmerken werde.

Die einzelnen Ports (Kanal 0 und Kanal 1):

Name	Port	Bezeichnung
CNTL0A	0	Control Register A Channel 0
CNTL0B	2	Control Register B Channel 0
STAT0	4	Status Register Channel 0
TDRO	6	Transmit Data Register Channel 0
RDRO	8	Receive Data Register Channel 0
CNTL1A	1	Control Register A Channel 1
CNTL1B	3	Control Register B Channel 1
STAT1	5	Status Register Channel 1
TDR1	7	Transmit Data Register Channel 1
RDR1	9	Receive Data Register Channel 1

Die einzelnen Bits der Ports:

CNTLA (Control A)

7	6	5	4	3	2	1	0
MPE	RE	TE	Kanal 0: RTS0*	Read: MPBR	MOD2	MOD1	MOD0
			Kanal 1: CKA1D	Write: EFR			

- MPE = Multiprocessor Mode Enable
- RE = Receiver Enable
- TE = Transmitter Enable
- RTS0* = Request To Send Channel 0 (aktiv LOW, nur Kanal 0!)
- CKA1D = Asynchronous Clock Channel 1 Disable (nur Kanal 1!)
- MPBR = Multiprocessor Bit Receive (nur beim Lesen!)
- EFR = Error Flag Reset (nur beim Schreiben!)
- MODx = ASCII Data Format Mode

Anmerkung: Bit 4 (D4) hat bei Kanal 0 eine andere Bedeutung als bei Kanal 1. Bit 3 (D3) bedeutet beim Schreiben etwas ganz anderes als beim Lesen.

CNTLB (Control B)

7	6	5	4	3	2	1	0
MPBT	MP	Read: CTS*	PEO	DR	SS2	SS1	SS0
		Write: PS					

- MPBT = Multiprocessor Bit Transmit
- MP = Multiprocessor Mode
- CTS = Clear To Send (nur beim Lesen!)
- PS = Prescale (nur beim Schreiben!)
- PEO = Parity Even/Odd
- DR = Divide Ratio
- SSx = Source/Speed Select

STAT (Status)

7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RIE	Kanal 0: DCDO*	TDRE	TIE
					Kanal 1: CTS1E		

- RDRF = Receive Data Register Full (nur lesbar)
- OVRN = Overrun Error (nur lesbar)
- PE = Parity Error (nur lesbar)
- FE = Framing Error (nur lesbar)
- RIE = Receive Interrupt Enable
- DCDO* = Data Carrier Detect (aktiv LOW, nur lesbar, nur Kanal 0!)
- CTS1E = Clear To Send Channel 1 Enable (nur Kanal 1!)
- TDRE = Transmit Data Register Empty (nur lesbar)
- TIE = Transmit Interrupt Enable

Einstellung der Kanäle

Über CNTLB wird die Baudrate, der Multiprozessor-Mode und die Parität festgelegt. D0 bis D3 und D5 sind für die Baudrate zuständig. Sie legen fest, durch welche Werte die Taktfrequenz geteilt wird.

D5	1.Teiler	D3	2.Teiler	D2	D1	D0	3.Teiler
0	10	0	16	0	0	0	1
1	30	1	64	0	0	1	2
				0	1	0	4
				0	1	1	8
				1	0	0	16
				1	0	1	32
				1	1	0	64
				1	1	1	externer Takt!

Der externe Takt wird von den Pins CKA0 (gemultipliziert mit DREQ0*) bzw. CKA1 (gemultipliziert mit TEND0*) abgenommen, wobei sich die Baudrate aus dem externen Takt geteilt durch 16 oder 64 (abhängig von D3) ergibt. Dies ist der Reset-Zustand. Falls die Baudrate vom System-Takt abgeleitet wird, wird der resultierende Takt über CKA0/1 ausgegeben!

Zwei Rechenbeispiele (bei einem Systemtakt von 9.216 Mhz):
 19200 Baud: $19200 = 9.216.000 \text{ DIV } 30 \text{ DIV } 16 \text{ DIV } 1 \Rightarrow \text{D5-D0} = 1x0000$
 300 Baud: $300 = 9.216.000 \text{ DIV } 30 \text{ DIV } 64 \text{ DIV } 16 \Rightarrow \text{D5-D0} = 1x1100$

D4 ist für die Art der Parität zuständig, falls Paritäts-Prüfung gewünscht wird: D4=0: gerade Parität (EVEN); D4=1: ungerade Parität (ODD).

17

Zu guter Letzt D6 und D7, die für den Multiprozessor-Modus zuständig sind: Wenn D6=1, dann wird statt des Paritäts-Bits ein Multiprozessor-Bit gesendet/empfangen. Wenn D7=1, wird beim Senden das Multiprozessor-Bit auf 1 gesetzt. Das könnte z.B. bedeuten: "Achtung, an alle anderen Prozessoren, es kommt was für alle!"

Weitere Einstellungen erfolgen über CNTLA. Mit D2 bis D0 wird das Datenformat festgelegt:

Bit	bedeutet
D2=0	7-Bit-Daten
=1	8-Bit-Daten
D1=0	keine Parität
=1	Parität
D0=0	1 Stop-Bit
=1	2 Stop-Bits

Wenn D3 mit 0 beschrieben wird, werden alle Fehler-Meldungen für diesen Kanal (im Register STAT0/1) gelöscht. Mit D4 kann bei Kanal 0 der Pin RTS0* gesetzt/gelöscht werden (Request To Send = Gegenseite darf senden). Bei Kanal 1 hingegen erfolgt hier die Auswahl, ob der Pin CKAL/TENDO* als Takteingang (CKA1, D4=0) oder als Kennung für das Ende eines Blocktransfers (TENDO*, D4=1) benutzt wird.

Mit D5 wird der Sender ein/ausgeschaltet, mit D6 der Empfänger. D7 hat nur im Multiprozessor-Modus eine Funktion. Wenn D7=1, werden nur Daten mit gesetztem MPB-Bit (s. D7 in CNTLB) angenommen, z.B. im Modus: "Empfange nur Daten für alle Prozessoren!". Bei D7=0 werden alle Daten angenommen. So ist folgende Kommunikation denkbar: Der Master-Prozessor sendet an alle Prozessoren (durch gesetztes MPB-Bit gekennzeichnet), welche beiden Prozessoren sich unterhalten dürfen. Diese schalten dann D7 auf 0 und wickeln ihre Kommunikation ohne MPB-Bit ab. Dann schalten sie D7 auf 1 und der Master-Prozessor wird benachrichtigt, die Sache wieder zu übernehmen.

Zu guter Letzt noch die Einstellungen, die über das Status-Register STAT vorgenommen werden. D0 legt fest, ob ein Interrupt erfolgt, wenn der Kanal sendebereit ist. Wenn D3=1, wird auch ein Interrupt ausgelöst, wenn ein Zeichen empfangen wurde oder ein Fehler auftrat. Bei Kanal 1 kann über D2 angewählt werden, ob der Pin CTS1*/RXS als CTS* (D2=1, Clear To Send = HD64180 darf senden) oder als Daten-Empfangs-Pin für CSI/O (D2=0) dient.

Senden/Empfangen

Der Status des Kanals kann über STAT abgefragt werden. D1=1 bedeutet, daß Daten gesendet werden dürfen. Wenn dann ein Byte in das Senderegister TDR geschrieben wird, geht D1 auf 0, bis das Zeichen gesendet wurde.

D7=1 bedeutet, daß ein Zeichen empfangen wurde und jetzt aus dem Empfangsregister RDR gelesen werden kann oder daß beim Empfangen ein Fehler aufgetreten ist. Nach dem Lesen des Bytes aus RDR wird D7 gelöscht. Folgende Fehler werden erkannt:

- D6 Overrun (letztes Byte wurde nicht schnell genug aus RDR geholt)
- D5 Parity (das Paritäts-Bit des Zeichens war falsch)
- D4 Framing (das/die Stop-Bits stimmten nicht)

Falls ein Fehler auftrat, muß dieser gelöscht werden, indem D3 von CNTLA mit 0 beschrieben wird. Achtung: dabei werden dann auch die anderen Einstellungen von CNTLA geändert!

18

Über D2 aus STAT kann bei Kanal 0 noch der Zustand des Signals "Data Carrier Detect" abgefragt werden. D2 wird auf 1 gesetzt, wenn das Signal am Pin DCDO* irgendwann mal HIGH war oder noch ist. D2 wird nach dem Lesen auf 0 gesetzt, wenn inzwischen DCDO*=LOW ist. Solange D2=1, wird nichts empfangen. Diese Verbindung ist nur bei bestimmten Modems sinnvoll.

Über D3 aus CNTLA kann das Multiprozessor-Bit des letzten empfangenen Bytes abgefragt werden. D5 aus CNTLB gibt Auskunft über den Zustand des Pins CTS* Für Kanal 0 gilt, daß der Kanal erst sendebereit wird (STAT0, D1=1), wenn der Pin CTS0*=0. CTS1* kann nur abgefragt werden und hat nur einen Einfluß auf die Sendebereitschaft, wenn wenn D2 aus STAT1 auf 1 gesetzt wurde (CTS1E = CTS-Enable Kanal 1).

Clocked Serial Input/Output

Diese Schnittstelle dient zur Verbindung mehrerer Prozessoren, wobei die anderen Prozessoren nicht unbedingt HD64180s sein müssen. Aber sie müssen über eine gemeinsame (synchronisierte) Taktquelle verfügen und die Send-/Empfangs-Operation muß in beiden Prozessoren gleichzeitig beginnen!

Der HD64180 kann nicht gleichzeitig empfangen und senden. Es gibt nur ein Send-/Empfangsregister: TRDR (Transmit/Receive Data Register), Port OBh. Es sind auch nur wenige Einstellungen möglich, so daß nur ein Kontrollregister nötig ist: CNTR (Control), Port OAh. Der Aufbau dieses Registers:

CNTR (Control)

7	6	5	4	3	2	1	0
EF	EIE	RE	TE	-	SS2	SS1	SS0

- EF = End Flag (nur lesbar)
- EIE = End Interrupt Enable
- RE = Receive Enable
- TE = Transmit Enable
- SSx = Speed Select

Einstellungen

Es kann nur die Baudrate eingestellt werden. Wenn die Baudrate neu gesetzt wird, sollte nicht gleichzeitig ein Send-/Empfangs-Zyklus (s.u.) gestartet werden. Der System-Takt wird durch den folgenden Wert geteilt (hier für 9.216 Mhz):

SS2	SS1	SS0	Teiler	ergibt Baudrate
0	0	0	20	460800
0	0	1	40	230400
0	1	0	80	115200
0	1	1	160	57600
1	0	0	320	28800
1	0	1	640	14400
1	1	0	1280	7200
1	1	1	externer Takt! (kleiner als 460800)	

Bei 111 wird der externe Takt über den Pin CKS eingelesen; ansonsten wird der resultierende Takt hier ausgegeben.

Empfangen/Senden

Zuerst muß abgefragt werden, ob der Sender gerade aktiv ist. Erst wenn D7 aus CNTR gesetzt ist, kann die Operation beginnen. Zum Senden wird das Byte in TRDR geschrieben und dann D4 in CNTR gesetzt. Gleichzeitig wird die Baudrate (wieder) gesetzt. Sie sollte aber nicht verändert werden, da es sonst Sende-Probleme geben kann. Die Operation ist erst beendet, wenn D7 wieder auf 1 geht. Falls D6 gesetzt wurde, wird dann ein Interrupt ausgelöst.

Beim Empfangen wird D5 gesetzt (zusammen mit der Baudrate). Nun beginnt der Empfangs-Vorgang, der immer gleich lang dauert. Wenn D7 wieder auf 1 geht, kann das Byte aus TRDR gelesen werden. Wenn vor dem Empfangen D6 gesetzt wurde, wird ein Interrupt ausgelöst, sobald D7 auf 1 springt. Es ist zu beachten, daß der Empfangs-Pin (RXS) mit CTS* für Kanal 1 gemultipliziert ist. Vor dem Empfang muß also über D2 von STAT1 die Funktion "Pin RXS/CTS1* als RXS" ausgewählt werden.

19 TIF wird erst gelöscht, wenn zuerst (!) TCR und dann TMDR (in der Reihenfolge Low-/High-Byte) gelesen werden. Ansonsten bleibt es gesetzt, was bei der Rückkehr von einer Interrupt-Routine sofort einen neuen Interrupt verursacht!

Die Timer des HD64180 von Gerald Schröder

Der HD64180 verfügt über zwei programmierbare 16-Bit-Zähler und einen nicht-programmierbaren 8-Bit-Zähler. Der 8-Bit-Zähler wird alle 10 Takte dekrementiert und kann nur gelesen werden, und zwar über Port 18h. Das Beschreiben dieses Ports kann Fehler im Timing des HD64180 verursachen.

Die 16-Bit-Zähler werden alle 20 Takte dekrementiert. Einer der 16-Bit-Zähler kann über den Pin TOUT an einen anderen Chip melden, daß er den Zählerstand 0000 erreicht hat. Beide Zähler können einen Interrupt auslösen, wenn sie 0000 erreichen.

Die Ports

Port	Abk.	Name
0Ch	TMDROL	Timer Data Register Channel 0 Low
0Dh	TMDROH	Timer Data Register Channel 0 High
0Eh	RLDR0L	Reload Data Register Channel 0 Low
0Fh	RLDR0H	Reload Data Register Channel 0 High
10h	TCR	Timer Control Register
14h	TMDR1L	Timer Data Register Channel 1 Low
15h	TMDR1H	Timer Data Register Channel 1 High
16h	RLDR1L	Reload Data Register Channel 1 Low
17h	RLDR1H	Reload Data Register Channel 1 High

TCR (Control)

0	1	2	3	4	5	6	7
TDE0	TDE1	TOC0	TOC1	TIE0	TIE1	TIFO	TIF1

TDE0/1 Timer Decrement Enable Channel 0/1
TOC0,1 Timer Output Control (beide für Kanal 1!)
TIE0/1 Timer Interrupt Enable Channel 0/1
TIFO/1 Timer Interrupt Flag Channel 0/1

Programmierung

Zuerst sollte der Timer gestoppt werden indem TDE in TCR auf 0 gesetzt wird. Dann das gewünschte Zählintervall in RLDR und (!) TMDR eintragen; die Eintragung in TMDR ist nötig, weil der Zähler sonst bei dem alten Wert startet. Es handelt sich hier um 16-Bit-Werte, so daß bei einem Takt von 9,216 Mhz Warteschleifen von ca. 2,2 Microsekunden bis ca. 0,142 Sekunden realisiert werden können; bzw. der Zähler erreicht 460800 (Intervall 0001) bis 7,03125 (Intervall FFFFh) Mal in der Sekunde den Stand 0000.

Der Zähler fängt erst an zu zählen, wenn TDE in TCR auf 1 gesetzt wird. Gleichzeitig wird über TIE ausgewählt, ob ein Interrupt ausgelöst werden soll, wenn TMDR den Zählerstand 0000 erreicht. Für Kanal 1 kann über TOC noch ausgewählt werden, welche Reaktion am Pin TOUT (Timer Output) beim Erreichen von 0000 erfolgt (Achtung: TOUT ist mit Adreßbit A18 gemultipliziert):

TOC0	TOC1	Reaktion TOUT
0	0	keine (Pin als A18)
0	1	TOUT wird getoggelt (0 -> 1, 1 -> 0)
1	0	TOUT auf 0
1	1	TOUT auf 1

Helmut Bernhardt

21

Wer in seinem Computer den Z80 durch einen HD64180 ersetzt hat, wird sicher schon ein plötzliches Booten oder einen Absturz beim Aufruf von bestimmten Programmen bemerkt haben, die vorher mit dem Z80 einwandfrei zusammengearbeitet haben. Zu solchen Programmen zählen REZILOG unter CP/M oder ZEUS unter NEWDOS/80. Der Fehler liegt nicht am Prozessor sondern am Programm. Der Programmierer hat dort Befehle benutzt, die von Zilog nicht dokumentiert sind aber trotzdem auf dem Z80 zur Verfügung stehen. Es sind dies die Illegals mit Doppelbyte-Op-codes.

Hitachi hat nun andere Vorstellungen entwickelt, wie diese freien Opcodes verwendet werden können, und eine Reihe dieser Codes den im Anhang zusammengestellten Befehlen zugeordnet. Gegenüber den Illegals des Z80 sind das sinnvolle Ergänzungen, die der Architektur des HD64180 entgegenkommen und u.a. eine elegante Programmierung der internen Ports der CPU ermöglichen. Die ebenfalls hinzugekommene Multiplikation ist leider nur auf 8Bit-Faktoren beschränkt geblieben.

Der wichtigste Befehl ist OTIMR, der eine Anzahl von Bytes, die in B vorgegeben wird, ab einer Adresse im Speicher, auf die HL zeigt, an aufeinander folgende interne Zeropage-Ports ausgibt, wobei die Adresse des ersten Ports in C vorzugeben ist. Damit kann z.B. der DMA-Controller schnell mit Parametern für eine Übertragung versorgt werden.

In mc 1/87, S.74 wurde gezeigt, wie die Illegals durch normale Befehle ersetzt werden können. Danach kann man von Hand entsprechende Programme auf normale Befehle umstellen. Was der Normalverbraucher zu Fuß macht, läßt der Experte die CPU zur Laufzeit des Illegal-verseuchten Programms selbst machen. Eventuell schreibt Gerald dazu nochmal was.

Bei Erreichen eines unbekanntes Opcodes wird der höchstpriorisierte TRAP-Interrupt ausgeführt, der einem CALL 0000H gleichkommt. Er ist mit einem RESET vergleichbar, nur daß kein richtiges /RESET-Signal erfolgt, das die gesamte Hardware zurücksetzt. Dadurch wird auch kein Floppy-Controller in den Einschaltzustand zurückgesetzt und auch kein ausgeblendetes EPROM wieder eingeschaltet.

Bei 0000H liegt dann RAM vor. Dort kann bei der Initialisierung ein Sprung auf die TRAP-Service-Routine gepatcht werden. Diese Routine kann mit dem Stack und den Flags im TRAP-Register herausfinden, ob ein TRAP die Ursache des Aufrufs war, oder ob die Adresse 0000H durch Software aufgerufen wurde; bei einem Trap kann sie ermitteln, welches Byte im Programm den TRAP auslöste und dem entsprechend alternative normale Befehle abarbeiten, und anschließend wieder die Kontrolle an das unterbrochene Programm zurückgeben.

Wenn eine solche TRAP-Service-Routine nicht vorliegt, bedeutet der CALL 0000H einen Warmstart unter CP/M oder einen Absturz unter NEWDOS/80, weil die BOOT-Routine im ROM nicht den Floppy-Controller auf Single Density zurücksetzt (deshalb funktioniert beim GENIE auch der Befehl BOOT oder ein JP 0000H nicht, wenn mit einem Doubler gearbeitet wird; der TRS 80 decodiert die Adresse 0000H und erzeugt daraus ein /RESET-Signal).

Natürlich funktioniert der TRAP-Interrupt nur bei den Z80-Illegals, für die der HD64180 keine neuen Befehle eingeführt hat. Wenn ein illegaler Opcode auftaucht, der einem HD64180-Befehl entspricht, so wird dieser Befehl nach Hitachi's Vorstellung ordnungsgemäß ausgeführt. Der Erfolg ist dann aber nicht der, den sich der Illegal-Programmierer einst gedacht hat.

Zusätzliche Befehle des HD64180

Multiplikation

ED 4C	MULT	BC	;BC := B * C	17 Taktzyklen
ED 5C	MULT	DE	;DE := D * E	
ED 6C	MULT	HL	;HL := H * L	
ED 4C	MULT	SP	;SP := SPhigh * SPlow	

Logisches AND zum Setzen der Flags ohne die Register zu verändern

ED 04	TEST	B	; A and B	7 Taktzyklen
ED 0C	TEST	C	; A and C	
ED 14	TEST	D	; A and D	
ED 1C	TEST	E	; A and E	
ED 24	TEST	H	; A and H	
ED 2C	TEST	L	; A and L	
ED 3C	TEST	A	; A and A	
ED 34	TEST	(HL)	; A and (HL)	10 Taktzyklen
ED 64 nn	TEST	nn	; A and nn	9 Taktzyklen
ED 74 pp	TESTIO	pp	; (C) and pp	12 Taktzyklen

Zeropage I/O-Befehle

ED 00 pp	INO	B,(pp)	; B := (pp)	12 Taktzyklen
ED 08 pp	INO	C,(pp)	; C := (pp)	
ED 10 pp	INO	D,(pp)	; D := (pp)	
ED 18 pp	INO	E,(pp)	; E := (pp)	
ED 20 pp	INO	H,(pp)	; H := (pp)	
ED 28 pp	INO	L,(pp)	; L := (pp)	
ED 30 pp	INO	F,(pp)	; S,Z,P <- (pp)	
ED 38 pp	INO	A,(pp)	; A := (pp)	
ED 01 pp	OUTO	(pp),B	; (pp) := B	13 Taktzyklen
ED 09 pp	OUTO	(pp),C	; (pp) := C	
ED 11 pp	OUTO	(pp),D	; (pp) := D	
ED 19 pp	OUTO	(pp),E	; (pp) := E	
ED 21 pp	OUTO	(pp),H	; (pp) := H	
ED 29 pp	OUTO	(pp),L	; (pp) := L	
ED 39 pp	OUTO	(pp),A	; (pp) := A	

ED 83	OTIM	; (C):=(HL), HL:=HL+1, C:=C+1, B:=B-1	14 Taktzyklen
ED 93	OTIMR	; (C)..(C+B) := (HL)..(HL+B), B:=0 ; oder: (C):=(HL), C:=C+1, HL:=HL+1, B:=B-1 ; wiederhole bis B=0	B*16 Taktzyklen
ED 8B	OTDM	; (C):=(HL), HL:=HL-1, C:=C-1, B:=B-1	14 Taktzyklen
ED 9B	OTDMR	; (C)..(C-B) := (HL)..(HL-B), B:=0 ; oder: (C):=(HL), C:=C-1, HL:=HL-1, B:=B-1	B*16 Taktzyklen
ED 76	SLP	; Sleep: Einstellen aller Aktivitäten incl. ; RAM-Refresh, bis /RESET oder /INT eintritt	

22