

ATA-to-ISA Busadapter Grundlagen

Stand: 1.2 vom 19. 9. 06

1. Was ist zu unterstützen?

E-A-Hardware mit ISA-Schnittstelle soll auch von modernen PC-Plattformen unterstützt werden können. Das betrifft vor allem PC/104-Moduln, aber auch Steckkarten im herkömmlichen (AT-) Formfaktor.

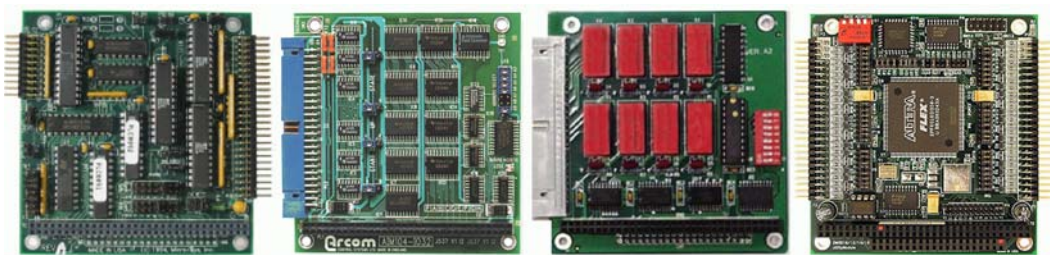


Abb. 1 Zu unterstützende Hardware (eine kleine Auswahl)

Typische Einsatzfälle:

- Weiternutzung vorhandener Hardware,
- Einsatz von PC/104-Moduln anstelle von Eigenentwicklungen (die Verfügbarkeit von PC/104 dürfte aller Voraussicht nach auch in den nächsten Jahren gegeben sein).

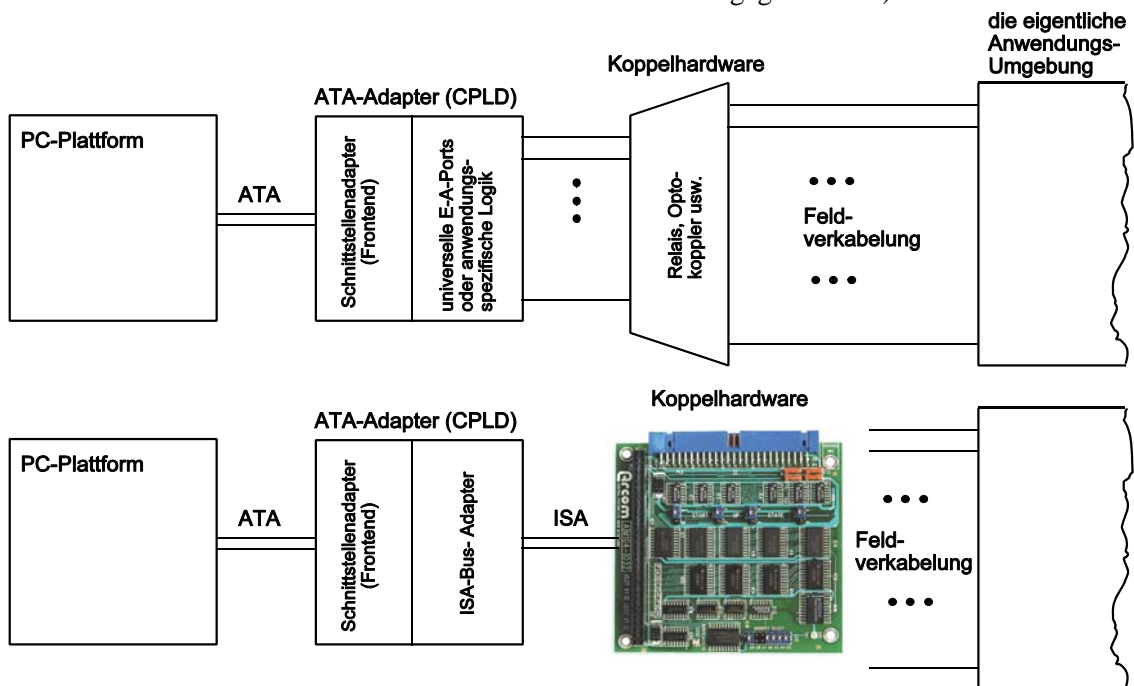


Abb. 2 Arbeitersparnis. Das Entwickeln von Platinen mit Relais, Optokopplern, A/D-Wandlern usw. ist meist nicht grundsätzlich schwierig, aber typischerweise kostspielig (wobei ggf. auch an aufwendige Tests, an Zertifizierungen usw. zu denken ist). Ein Ausweg: fertige PC/104-Moduln

Der wesentliche Vorteil:

Um z. B. PC/104-Moduln einsetzen zu können, ist man nicht auf (teure) Plattformen aus dem Bereich der Industrie-PCs angewiesen. Statt dessen können Motherboards aus der Massenfertigung eingesetzt werden. Das betrifft vor allem die kleinen Formfaktoren (MiniITX, NanoITX usw.). Durch Nutzung des ATA-Interfaces bleiben die modernen Schnittstellen dieser Boards (z. B. PCI oder PCI Express) für PC-typische Erweiterungen frei (Video, Vernetzung usw.) – es ist nicht erforderlich, den womöglich einzigen Erweiterungs-Slot zu belegen, um Relais zu schalten und Optokoppler abzufragen.

Der minimale Funktionsumfang

Derartige Funktionseinheiten sind meist nur für die einfachste Art der ISA-Zugriffe ausgelegt: 8-Bit-E-A-Zugriffe ohne Wartezustände. Ersichtlicherweise (Abb. 3) genügen ein 8-Bit-Datenbus und die Unterstützung der 10 niedrigstwertigen Adreßleitungen (entsprechend der 10-Bit-Adressierung bei E-A-Zugriffen im ursprünglichen IBM-PC (Plattform-E-A)).

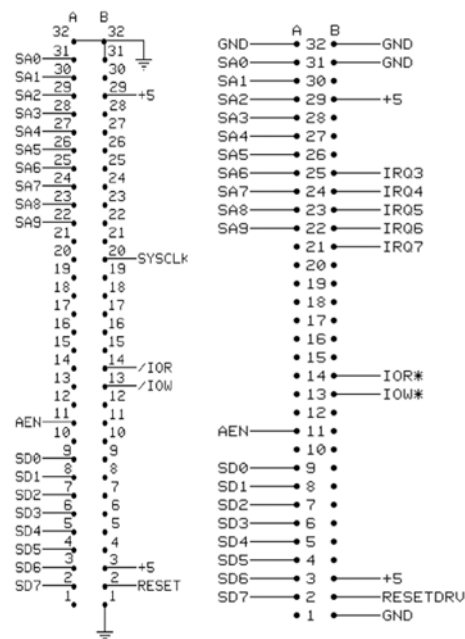


Abb. 3 Typische ISA-Signalbelegungen (Arcom)

Weitere Merkmale:*Interrupts*

Einige Funktionseinheiten können Interruptanforderungen stellen. Unterstützung durch Abfrage (Polling) oder durch Weitergabe auf die Interruptleitung der ATA-Schnittstelle (erfordert eigenen Interrupthandler im PC).

Takte

Manche Funktionseinheiten nutzen Taktsignale, die standardgemäß über den ISA-Bus geliefert werden. Hierzu sind ggf. entsprechende Taktgeneratoren vorzusehen (hat mit dem ATA-Adapter an sich nichts zu tun).

Rücksetzen

Das Rücksetzsignal der ATA-Schnittstelle wird zum ISA-Bus weitergereicht. Darüber hinaus sollte es möglich sein, das Rücksetzen am ISA-Bus auch programmseitig auszulösen.

Wieviele Funktionseinheiten am ISA-Bus?

Es ist unwahrscheinlich, daß gleichsam klassische, voll ausgebaute ISA-Konfigurationen zu unterstützen sind. Annahme: an einen ATA-Adapter werden typischerweise nicht mehr als vier Funktionseinheiten angeschlossen; oftmals wird es sich nur um eine oder um zwei handeln.

Adressierung

Da der ISA-Bus mit der PC-Plattform gar nichts zu tun hat, kann es auch keine Adreßkonflikte mit der Plattform-Hardware geben. Der Adreßraum der einzelnen Funktionseinheiten ist typischerweise klein (z. B. nur vier aufeinanderfolgende E-A-Adressen). Man könnte auch daran denken, weniger als 10 Adreßsignale vorzusehen (und die weggelassenen höherwertigen Adreßpositionen auf dem Bus ggf. mit Festwerten zu belegen).

Hinweise:

1. Wieviele Adreßbits zu unterstützen sind, hängt vor allem davon ab, wie die Adreßeinstellung in den Funktionseinheiten ausgelegt ist (manche sind nur im Rahmen fester Adreßbereiche konfigurierbar).
2. Es ist gelegentlich vorteilhaft, neben einem allgemeinen AEN-Signal gesonderte AEN-Signale für die einzelnen Steckpositionen anzubieten (geographische Adressierung).

Formfaktoren der ATA-Busadapter

Der Busadapterschaltkreis wird typischerweise auf einer Leiterplatte untergebracht, die auch die ISA-Steckverbinder aufnimmt. Die Abmessungen sind im Grunde beliebig. Es liegt aber nahe, industrieübliche Formfaktoren zu bevorzugen. Es kommen vor allem in Betracht:

- PC/104-Modul (Busadapter anstelle des CPU-Moduls im PC/104-Stapel),
- PC/104-Basisplatine, z. B. gemäß den Formfaktoren EBX (146 • 203 mm) oder EPIC (115 • 165 mm),
- Basisplatine, die an den ATX-Formfaktor angepaßt ist (so daß sie z. B. in einem üblichen PC-Gehäuse neben einem MiniITX-Motherboard angeordnet werden kann),
- Basisplatine in einem der Formfaktoren, die für miniaturisierte Industrie-PCs üblich sind (3½“, 5¼“ (Grundfläche wie die entsprechenden Laufwerke), Europakarte (100 • 160 mm) usw.).

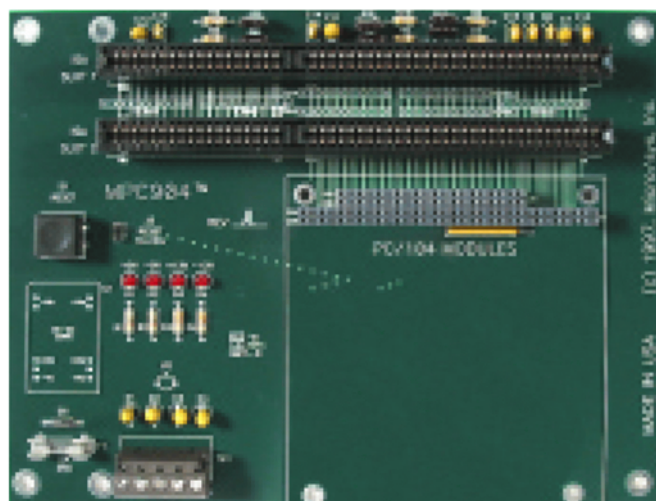
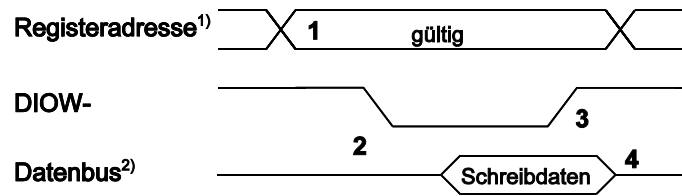


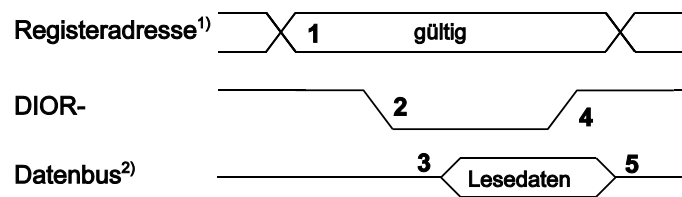
Abb. 4 Eine Basisplatine für ISA-Hardware mit PC/104-Steckplatz und zwei ISA-Slots (Microsys)

2. ATA Timing

a) Schreibzugriff



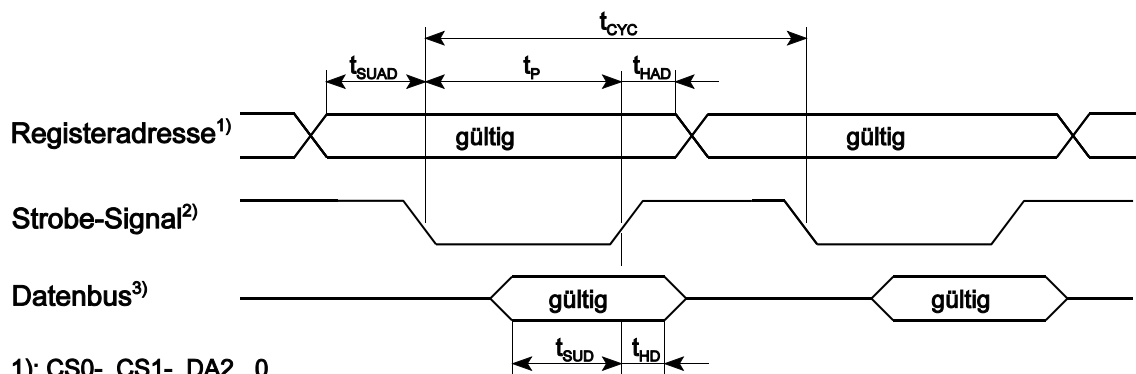
b) Lesezugriff



1): CS0-, CS1-, DA2...0
2): Datenbits DD7...0 (8-Bit-Zugriffe)

Abb. 5 ATA-Registerzugriffe (1). Schreiben und Lesen

- a) Schreibzugriff: 1 - Host schaltet Registeradresse auf; 2 - Host aktiviert das Strobesignal und belegt den Datenbus mit dem zu schreibenden Byte; 3 - Host deaktiviert das Strobesignal. Infolgedessen werden die Schreibdaten vom Gerät übernommen; 4 - Host gibt Datenbus frei.
- b) Lesezugriff: Der Datenbus ist anfänglich hochohmig. 1 - Host schaltet Registeradresse auf; 2 - Host aktiviert das Strobesignal; 3 - Gerät belegt den Datenbus mit dem gelesenen Byte; 4 - Host übernimmt die Lesedaten und deaktiviert das Strobesignal; 5 - Gerät gibt Datenbus frei.



1): CS0-, CS1-, DA2...0
2): Schreiben: DIOW-, Lesen: DIOR-
3): Datenbits DD7...0 (8-Bit-Zugriffe)

Abb. 6 ATA-Registerzugriffe (2). Der allgemeine Ablauf (ohne Wartezustände)

Kennwert	Bedeutung	Betriebsart (Mode)				
		0	1	2	3	4
t_{CYC}	Zykluszeit (schnellste Folge von Registerzugriffen)	600	383	330	180	120
t_p	minimale Strobe-Impulsdauer	290	290	290	80	70
t_{SUAD}	minimale Setup-Zeit für Registeradressierung	70	50	30	30	25
t_{HAD}	minimale Haltezeit für Registeradressierung	20	15	10	10	10
t_{SUD}	minimale Setup-Zeit für Datenbelegung (Schreiben)	60	45	30	30	20
	minimale Setup-Zeit für Datenbelegung (Lesen)	50	35	20	20	20
t_{HD}	minimale Haltezeit für Datenbelegung (Schreiben)	30	20	15	10	10
	minimale Haltezeit für Datenbelegung (Lesen)	5	5	5	5	5

Tabelle 1 Zeitkennwerte der ATA-Registerzugriffe (alle Angaben in ns)

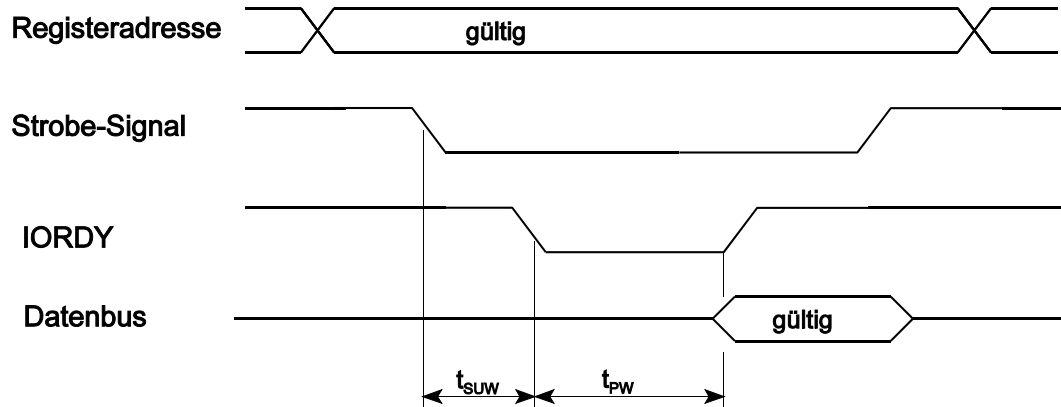


Abb. 7 ATA-Registerzugriffe (3). Vom Gerät verlängerter Zugriff (Einfügen von Wartezuständen)

Das Gerät kann durch Aktivieren von IORDY den jeweiligen Zugriff verlängern, also Wartezustände einfügen. IORDY muß spätestens 35 ns nach Aktivierung des Strobe-Signals erregt werden (t_{suw}) und darf höchstens 1,25 μ s lang aktiv bleiben (t_{pw}).

Kennwert	Bedeutung	Zeitangabe (in ns)
t_{suw}	maximale Setup-Zeit zum Einleiten des Wartezustandes	35
t_{pw}	maximale Dauer des Wartezustandes (IORDY-Impulsdauer)	1250

Tabelle 2 Zeitkennwerte zu Abb. 7

3. ISA Timing

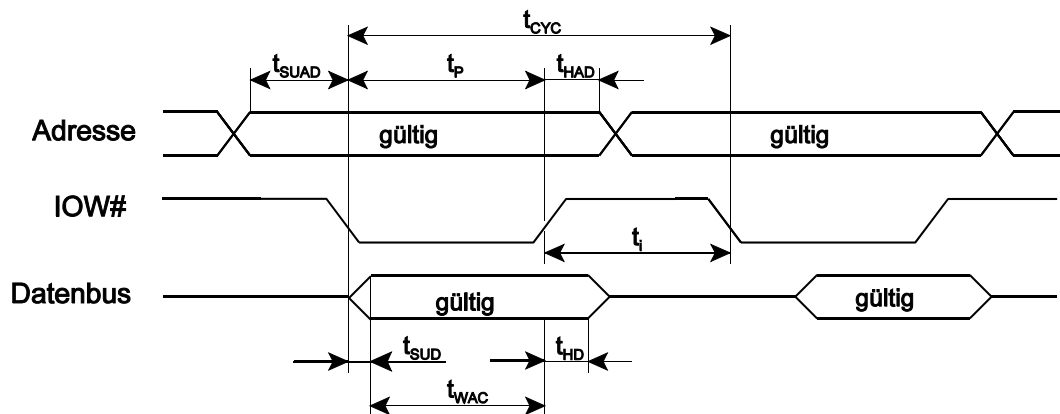


Abb. 8 ISA-E-A-Zugriffe (1). 8-Bit-Schreibzugriff (Ausgabe)

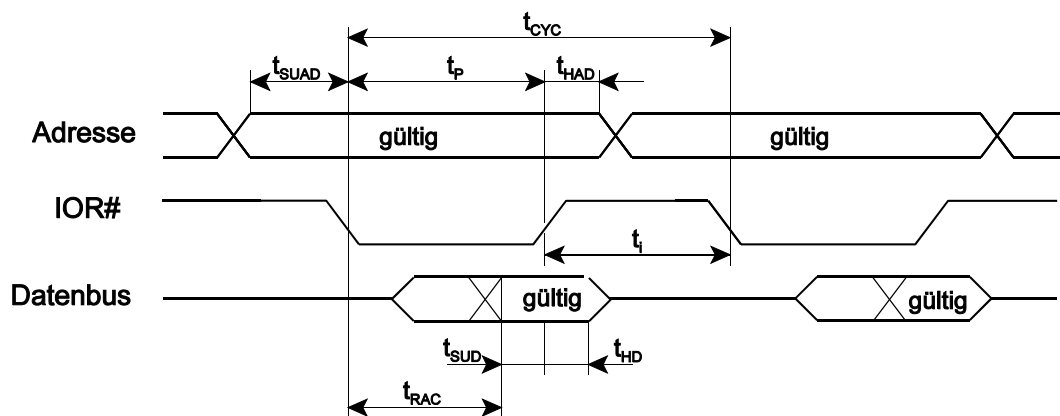


Abb. 9 ISA-E-A-Zugriffe (2). 8-Bit-Lesezugriff (Eingabe)

Kennwert	Bedeutung	Normalzugriff	Zugriff ohne Wartezustände
t_{CYC}	Zykluszeit (schnellste Folge von Buszugriffen) = $t_p + t_i$	683	312
t_p	minimale Strobe-Impulsdauer	520	149
t_i	minimale Pausendauer zwischen zwei Zugriffen	163	163
t_{SUAD}	minimale Setup-Zeit für Adresse	100	100
t_{HAD}	minimale Haltezeit für Adresse	41	41
t_{SUD}	minimale Setup-Zeit für Datenbelegung (Schreiben)	-40	-40
	minimale Setup-Zeit für Datenbelegung (Lesen) = $t_p - t_{RAC}$	38	24
t_{WAC}	minimale Datenzugriffszeit (Schreiben) = $t_p + t_{SUD}$	480	109

Kennwert	Bedeutung	Normalzugriff	Zugriff ohne Wartezustände
t_{RAC}	maximale Datenzugriffszeit (Lesen)	482	125
t_{HD}	minimale Haltezeit für Datenbelegung (Schreiben)	36	36
	minimale Haltezeit für Datenbelegung (Lesen)	0	0

Tabelle 3 Zeitkennwerte der ISA-E-A-Zugriffe (8 Bits)

4. Probleme

ATA war ursprünglich eine Art Verlängerung des ISA-Bus. Moderne ATA-Spezifikationen beziehen sich jedoch nicht mehr auf diese ursprünglichen Voraussetzungen; die ATA-Zeitspezifikationen weichen von denen des ISA-Bus teils erheblich ab (ATA-Timing \neq ISA-Timing).

Womit ist in der Praxis zu rechnen?

1. damit, daß sich die Entwickler der ISA-Hardware auf die Zeitpezifikationen dieses Bussystems verlassen haben. Das betrifft z. B. den Einsatz von LS-TTL-Logikschaltkreisen und die Nutzung von Latches (z. B. 74LS373 oder 573) als Datenregister.
2. damit, daß moderne ATA-Hostadapter zwar die ATA-Spezifikationen erfüllen, nicht aber die ISA-Spezifikationen.

Hinweis:

Traditionell haben die Motherboard-Schaltkreissätze die ISA-Zeitspezifikationen an der ATA-Schnittstelle eingehalten (ATA-Timing in PIO Mode 0 = ISA-Timing). Bei modernen Schaltkreissätzen (die keinen ISA-Bus, X-Bus oder dergleichen unterstützen) kann man sich jedoch nicht mehr darauf verlassen.

Schreiben über den ISA-Bus:

- der Schreibimpuls ist wenigstens 520 ns breit,
- bei aktivem Schreibimpuls werden die Daten mindestens 480 ns lang bereithalten,
- nach dem Deaktivieren des Schreibimpulses werden die Daten für wenigstens 36 ns weiterhin auf dem Bus gehalten.

Schreiben über die ATA-Schnittstelle (PIO Modes 0 bis 2):

- es kann sein, daß der Schreibimpuls nur 290 ns breit ist,
- die Daten können erst kurz vor dem Ende des Schreibimpulses gültig werden (Setup-Zeiten: 70, 50 oder 30 ns); sie sind also nicht wenigsten 480 ns lang gültig, sondern beispielsweise nur 70 ns (die Gültigkeit über nahezu die gesamte Schreibimpulsdauer ist nicht garantiert),
- nach dem Deaktivieren des Schreibimpulses können die Daten schon nach kurzer Zeit vom Bus verschwinden (Haltezeiten: 20, 15 oder 10 ns).

Lesen über den ISA-Bus:

- der Leseimpuls ist wenigstens 520 ns breit,
- Die Datenzugriffszeit (in der angesprochenen Einrichtung) darf bis zu 480 ns betragen, ohne daß Wartezustände eingeschoben werden müssen.

Lesen über die ATA-Schnittstelle (PIO Modes 0 bis 2):

- es kann sein, daß der Leseimpuls nur 290 ns breit ist,
- die Datenzugriffszeit (im Sinne von Leseimpulsdauer - Setup-Zeit) beträgt im PIO Mode 0 nur $290 - 50 = 240$ ns.

Entscheidung über das Einleiten von Wartezuständen:

Am ISA-Bus stehen dafür 80 ns zur Verfügung (bezogen auf die Vorderflanke des jeweiligen Strobe-Impulses), an der ATA-Schnittstelle (vgl. Tabelle 2) nur 35 ns.

Was passieren kann:

Manchmal sind die Zeiten, die die ATA-Schnittstelle einräumt, zu knapp, weil sich die Entwickler der ISA-Hardware auf die ISA-Spezifikation verlassen haben. Beispiele:

- Einsatz von Latches: der 74LS373 erfordert eine Haltezeit von wenigstens 20 ns.
- Schaltungen zum Erkennen von Wartezuständen (z. B. Adreßdecoder) brauchen länger als 35 ns.
- Zugriffe auf kostengünstige (= nicht übertrieben schnelle) SRAMs und FIFOs sind nicht in 70 ns zu erledigen, Zugriffe auf ebensolche ROMs nicht in 240 ns.
- die Steuersignale werden nicht direkt an den Flipflops wirksam, sondern laufen über Gatternetzwerke, Decoder usw. Im Beispiel von Abb. 10 werden keine Latches, sondern D-Flipflop-Register als Datenregister eingesetzt. Das Taktsignal am Register ist aber nicht direkt IOW#, sondern ein über den Schreibdecoder verzögerter Impuls. Wenn die Daten eher vom Bus verschwinden, als dieser Impuls am Register wirksam wird, funktioniert es nicht ...

Hinweis:

Daß ein solcher Problemfall vorliegt, äußert sich in der Praxis vor allem durch zeitweilige Fehler (die wie gelegentliche Aussetzer aussehen) oder durch Probleme bei Übergang zu anderer Hardware (Beispiel: Einsatz eines PC/104-Moduls eines anderen Anbieters).

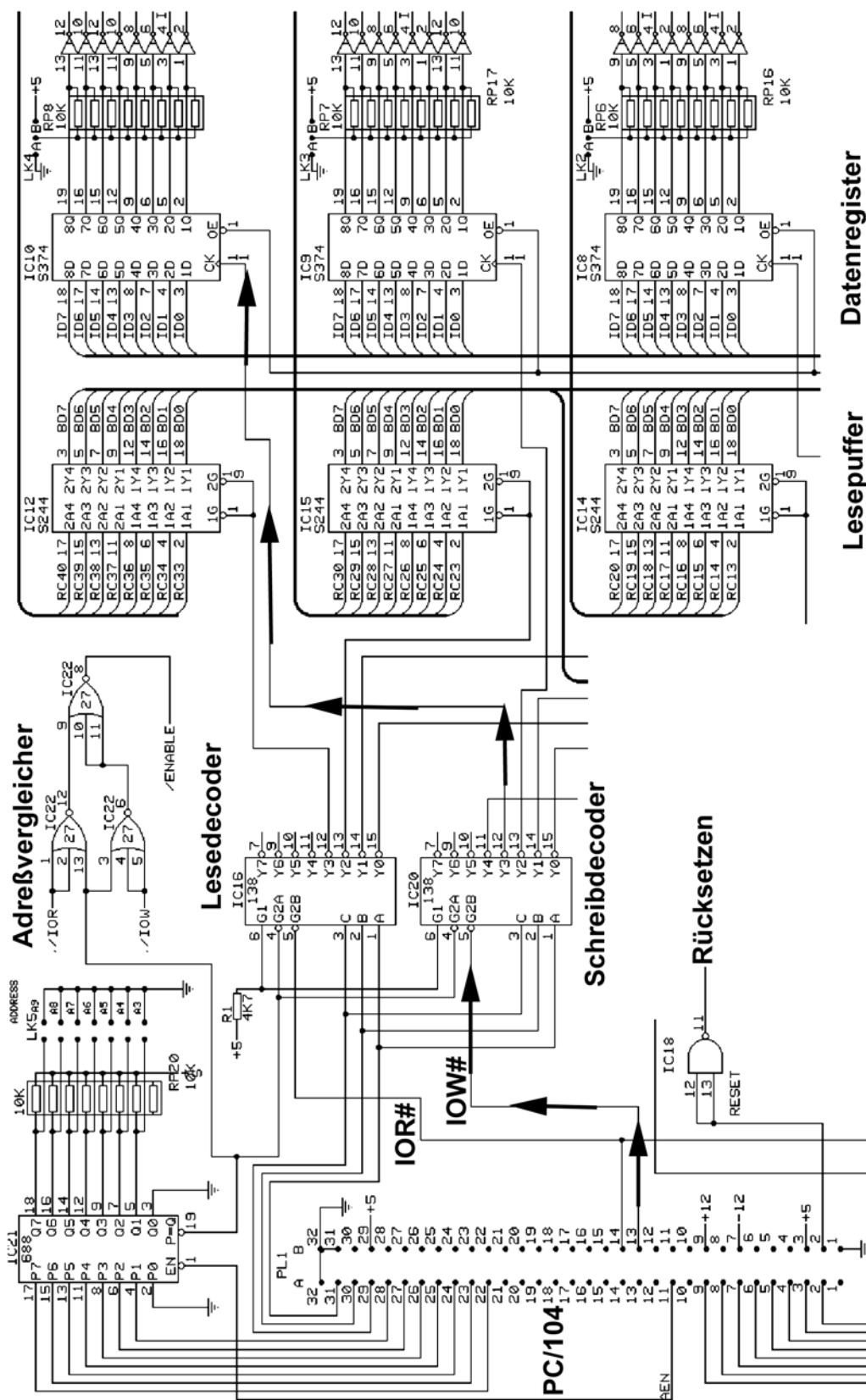


Abb. 10 Ausschnitt aus einer Praxisschaltung (Arcom). Die Pfeile veranschaulichen den Fluß der Schreibtaktimpulse

5. Lösungsansätze

1. Emulation ISA-gerechter Zeitverhältnisse durch Folgen von ATA-Zugriffen

Für jeden ISA-Zugriff werden zwei ATA-Zugriffe ausgeführt. Die Zeitverhältnisse ergeben sich aus dem Abstand der Zugriffe und der Breite der ATA-Strobesignale.

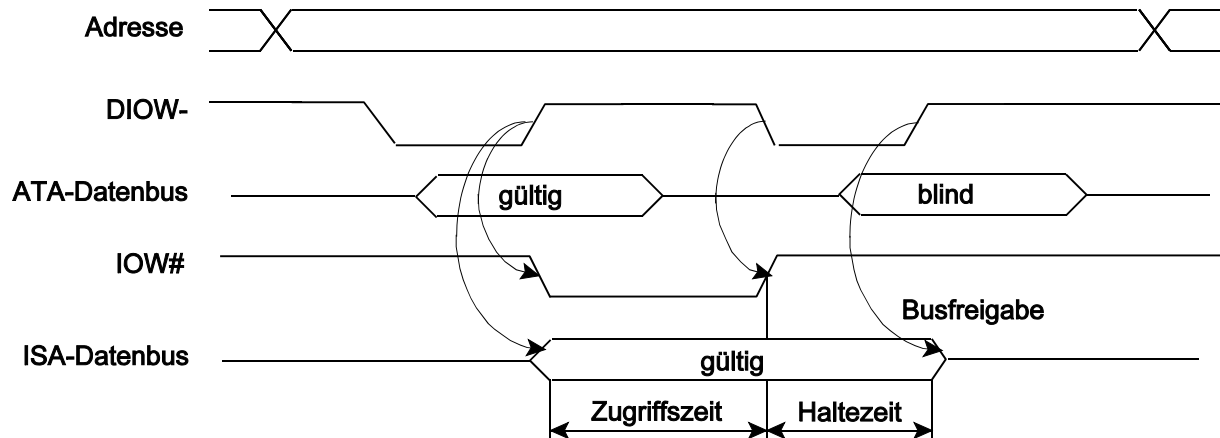


Abb. 11 Emulation eines ISA-Schreibzugriffs

Der 1. ATA-Zugriff liefert die Schreibdaten und startet den Schreibzyklus. Die Rückflanke (Low-High-Flanke) von DIOW- veranlaßt das Einschalten von IOW# und das Aufschalten der Daten auf den ISA-Bus. *Hinweis:* Zur Zeit der Low-High-Flanke von DIOW- sind die ATA-Schreibdaten garantiert im Adapter angekommen (auch bei geringer Setup-Zeit (z. B. 30 ns in PIO Mode 2)).

Der 2. ATA-Zugriff schaltet den Schreibzyklus wieder aus und nimmt etwas später (Haltezeit) die Schreibdaten vom ISA-Bus. Die Vorderflanke (High-Low-Flanke) von DIOW- veranlaßt das Ausschalten von IOW#, die Rückflanke (Low-High-Flanke) veranlaßt die Busfreigabe.

Die Dauer des ISA-Zugriffs wird durch den Abstand zwischen den ATA-Zugriffen bestimmt, die Haltezeit auf dem ISA-Datenbus durch die Breite des DIOW-Impulses.

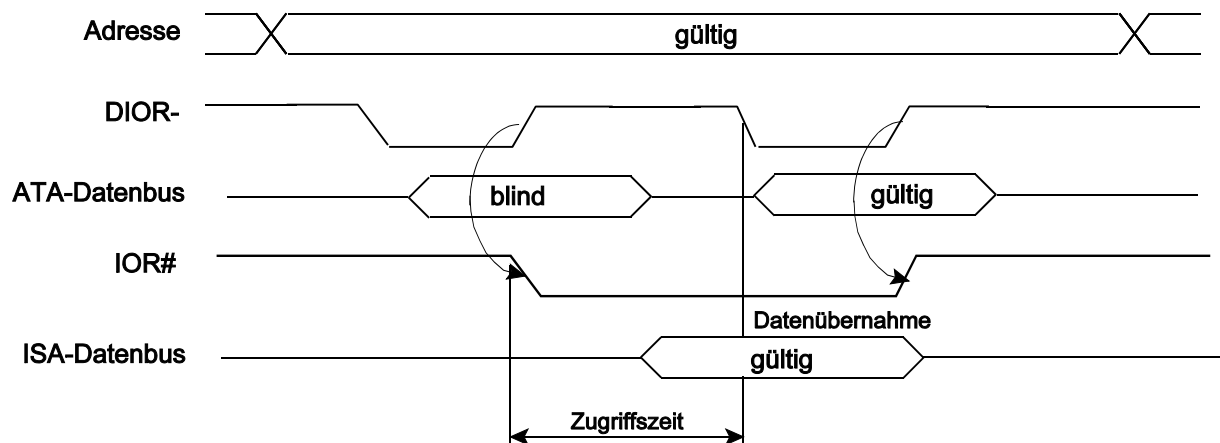


Abb. 12 Emulation eines ISA-Lesezugriffs

Der 1. ATA-Zugriff ist ein blinder Lesezugriff, der lediglich das IOR#-Signal des ISA-Bus einschaltet.

Beim 2. ATA-Zugriff werden die gelesenen Daten mit der Vorderflanke (High-Low-Flanke) von DIOR- vom ISA-Bus in den ATA-Adapter übernommen. Die Rückflanke (Low-High-Flanke) von DIOR- veranlaßt das Ausschalten von IOR# .

Die Dauer des ISA-Zugriffs – und damit die in der ISA-Einrichtung verfügbare Zugriffszeit – wird durch den Abstand zwischen den ATA-Zugriffen bestimmt.

Vorteile:

- keine zeitbestimmenden Glieder im ATA-Adapter erforderlich,
- ISA-Zyklen mit bei weitem ausreichenden Zeitreserven.

Nachteile:

- vergleichsweise geringe Datenraten,
- ISA-Zyklen können extrem lang werden (viele Millisekunden), wenn das Betriebssystem dem Programm zwischen den beiden Zugriffen eines ISA-Zyklus die Laufzeit entzieht. Abhilfe: komplette Zugriffsfunktionen als eigenständige Treiberroutinen implementieren (ein ISA-Zugriff = zwei ATA-Zugriffe, die im Kernel Mode des Systems unteilbar hintereinander ausgeführt werden).

2. Sequentielle Ablaufsteuerung im ATA-Adapter

Schreiben:

Der ATA-Zugriff liefert die Schreibdaten zum ATA-Adapter. Die Rückflanke (Low-High-Flanke) von DIOW- veranlaßt den Ablauf des ISA-Schreibzyklus, der von einer Steuerschaltung autonom ausgeführt wird.

Lesen:

Der Lesezugriff wird zum ISA-Bus weitergereicht (IOR# entspricht DIOR-), der ATA-Zyklus wird aber durch Wartezustände entsprechend verlängert. Die Steuerschaltung bestimmt die Dauer des Wartens und damit des ISA-Zyklus.