LCD-Anzeigen mit Parallelinterface

Das Interface besteht typischerweise aus einem 8-Bit-Datenbus und einigen Steuerleitungen. Es kann rein softwareseitig angesteuert werden (programmierte Ein- und Ausgabe über universelle Ports). Manche Anzeigen lassen sich in einem 4-Bit-Modus betreiben.

Dotmatrixanzeigen
Diese Anzeigen (Abb. 1 und 2) können lediglich Zeichen aus einem bestimmten Zeichenvorrat darstellen. Die Anzeigekapazität ist durch die Anzahl der Zeichen je Zeile und die Anzahl der Zeilen gegeben.

Graphische Anzeigen
Diese Anzeigen (Abb. 3 und 4) haben ein durchgehendes Pixelraster, wobei jeder einzelne Bildpunkt programmseitig zugänglich ist. Somit lassen sich beliebige graphische Darstellungen aufbauen. Die Anzeigekapazität ist durch die Anzahl der Bildpunkte (Pixel) in waagerechter und senkrechter Richtung gegeben.

Die Schnittstellen sind Industriestandards
Die Schnittstelle einer LCD-Anzeige wird vor allem durch die eingebauten Controller bestimmt. Einige Typen sind Industriestandards.

Der Funktionsvielfalt sind keine Grenzen gesetzt
Es gibt Anzeigen mit wählbaren Festtexten, mit Kommandosätzen zum Erzeugen graphischer Darstellungen, mit mehreren Zeichensätzen usw. – und es erscheint immer wieder Neues. Im folgenden wollen wir uns jedoch auf die einfachsten (= kostengünstigsten) Typen beschränken.

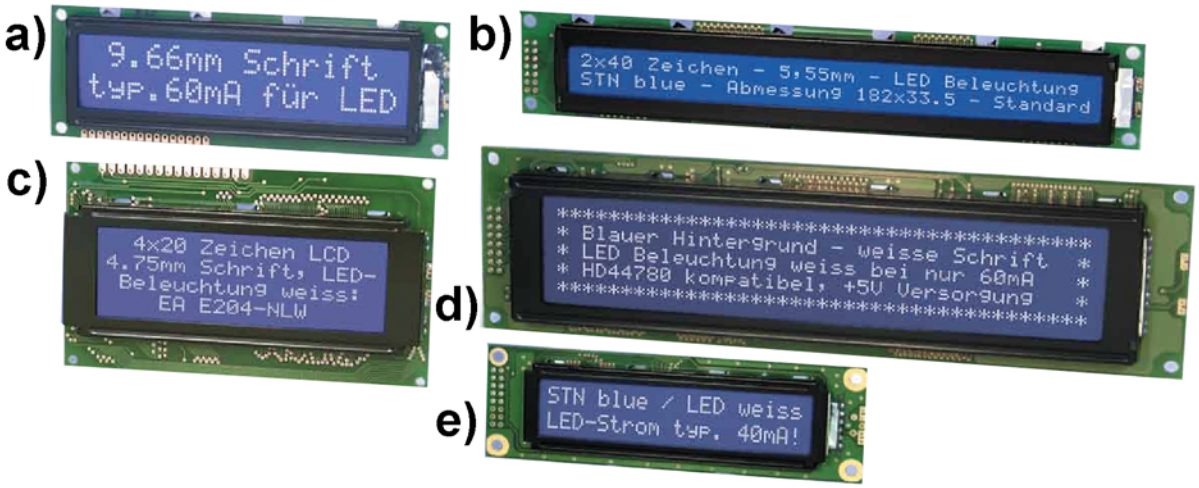

Abb. 1 Dotmatrixanzeigen (1). Herkömmliche Auslegungen (Electronic Assembly). Schraubbefestigung. a) - 2 Zeilen zu 16 Zeichen; b) - 2 Zeilen zu 40 Zeichen; c) - 4 Zeilen zu 20 Zeichen; d) - 4 Zeilen zu 40 Zeichen; e) - 2 Zeilen zu 20 Zeichen

Die Anschlußanordnungen bilden eine Art Industriestandard: 14 Anschlüsse für Signale und Spannungsversorgung, 2 Anschlüsse für die Hintergrundbeleuchtung. Typische Anordnungen: in einer Reihe unten (a), in einer Reihe oben (c), in zwei Reihen links (b, d, e). Zwischen verschiedenen Modellen gibt es – auch bei gleicher Darstellkapazität – gelegentlich spitzfindige Unterschiede in der mechanischen Auslegung. Kataloge genau studieren!

Abb. 2 Dotmatrixanzeigen (2). Ein alternativer Formfaktor (Electronic Assembly). Zum Aufstecken/Aufschnappen

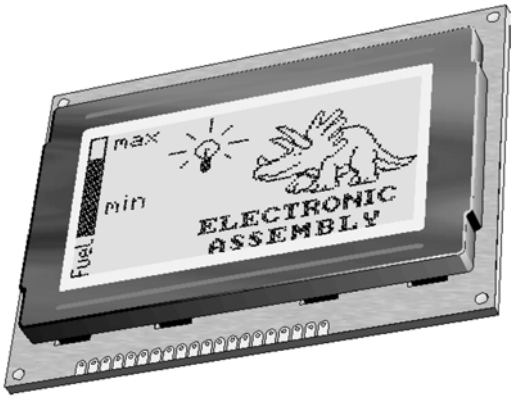

Abb. 3 Graphikanzeige (1). Ein Einfachmodell (Electronic Assembly). Kann nur Punktraster darstellen (Darstellung ist punktweise auszuprogrammieren). Darstellkapazität z. B. 128 • 64 Bildpunkte

Abb. 4 Graphikanzeige (2). Kann Graphik und mehrere Zeichensätze darstellen (Electronic Assembly)

1. Dotmatrixanzeigen

Die meisten Dotmatrixanzeigen sind mit Display-Controllern HD44780 (Hitachi) oder kompatiblen Schaltkreisen bestückt. Somit ergibt sich eine typische Auslegung der Schnittstelle mit 8-Bit-Datenbus und 3 Steuersignalen (Abb. 5). Die Schnittstelle kann in einem 4-Bit-Modus betrieben werden (Abb. 6).

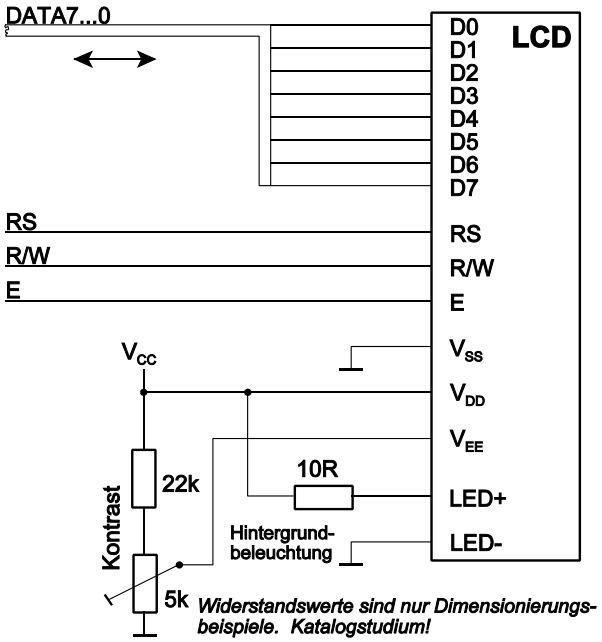

Abb. 5 Dotmatrixanzeige mit 8-Bit-Interface

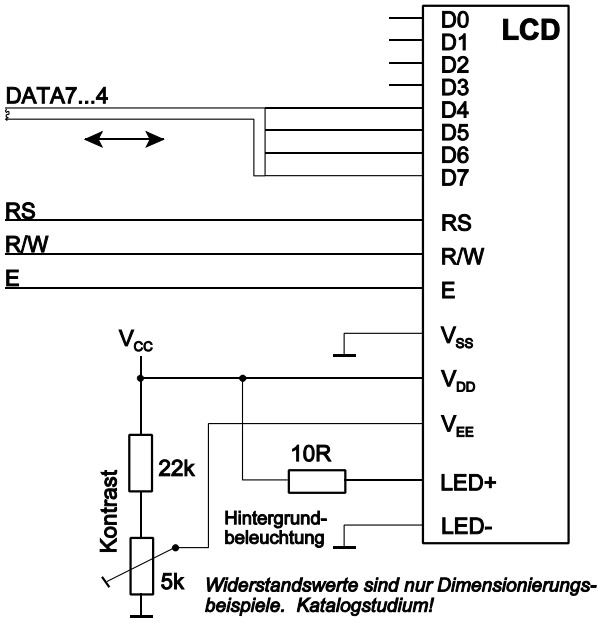

Abb. 6 Dotmatrixanzeige mit 4-Bit-Interface

1.1 Interfacesignale im Überblick

DATA7...0
Bidirektionaler Datenbus. Zur Übertragung von Daten, Adressen, Kommandos und Zustandsmeldungen.

Der inaktive Datenbus
Ist der Datenbus nicht aktiv, so wird er auf High-Pegel gezogen (über Feldeffekttransistoren, die als Pull-up-Widerstände wirken). Bereich der Widerstandswerte: 25...300 kΩ (typisch 60 kΩ).

Der Datenbus im 4-Bit-Betrieb
Es werden nur die Bitpositionen 7...4 genutzt. Die Bitpositionen 3..0 dürfen offengelassen werden.

RS: Register Select
Eingang zur Registerauswahl:
- RS = 0: Kommando- oder Zustandsübertragung,
- RS = 1: Datenübertragung.

R/W: Read / Write
Eingang zur Lese-Schreib-Steuerung:
- R/W = 0: Schreibzugriff. Bus wird vom Controller nicht aktiviert (kann also mit zu schreibenden Daten belegt werden).
- R/W = 1: Lesezugriff. Wenn E = 1, wird Bus vom Controller aktiviert und mit Lesedaten belegt.

E: Enable
Eingang. Allgemeines Erlaubnissignal:
- E = 0: keine Wirkung. Controller inaktiv (nicht ausgewählt). Datenbus frei.
- E = 1: Wirkung gemäß Belegung von RS und R/W.

1.2 Signalspiele am Interface

Es gibt im Grunde nur zwei Zugriffsarten: Schreiben und Lesen (Abb. 7 und 8, Tabellen 1 und 2). Alles weitere hängt vom Auswahlsignal RS und (beim Schreiben) von der Belegung des Datenbytes ab.

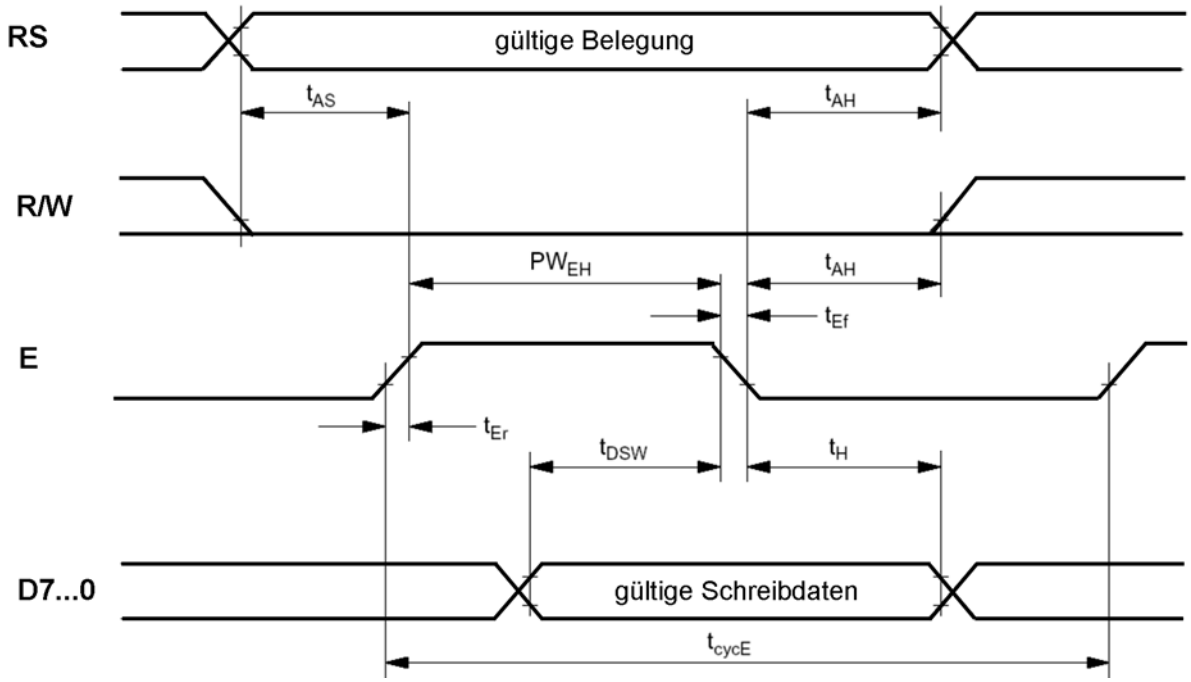

Abb. 7 Der Ablauf eines Schreibzugriffs

Kennwert	Symbol	HD44780 o. kompatibel			HD44780U		
		min.	typ.	max.	min.	typ.	max.
ENABLE-Zykluszeit	t_{cycE}	1000	-	-	500	-	-
ENABLE-Impulsdauer	PW_{EH}	450	-	-	230	-	-
ENABLE-Anstiegs- und Abfallzeit	t_{Er}, t_{Ef}	-	-	25	-	-	20
Adressen-Vorhaltezeit	t_{AS}	140	-	-	40	-	-
Adressen-Haltezeit	t_{AH}	10	-	-	10	-	-
Daten-Vorhaltezeit	t_{DSW}	195	-	-	80	-	-
Daten-Haltezeit	t_H	10	-	-	10	-	-

Tabelle 1 Zeitkennwerte des Schreibzugriffs. Alle Zeitangaben in ns

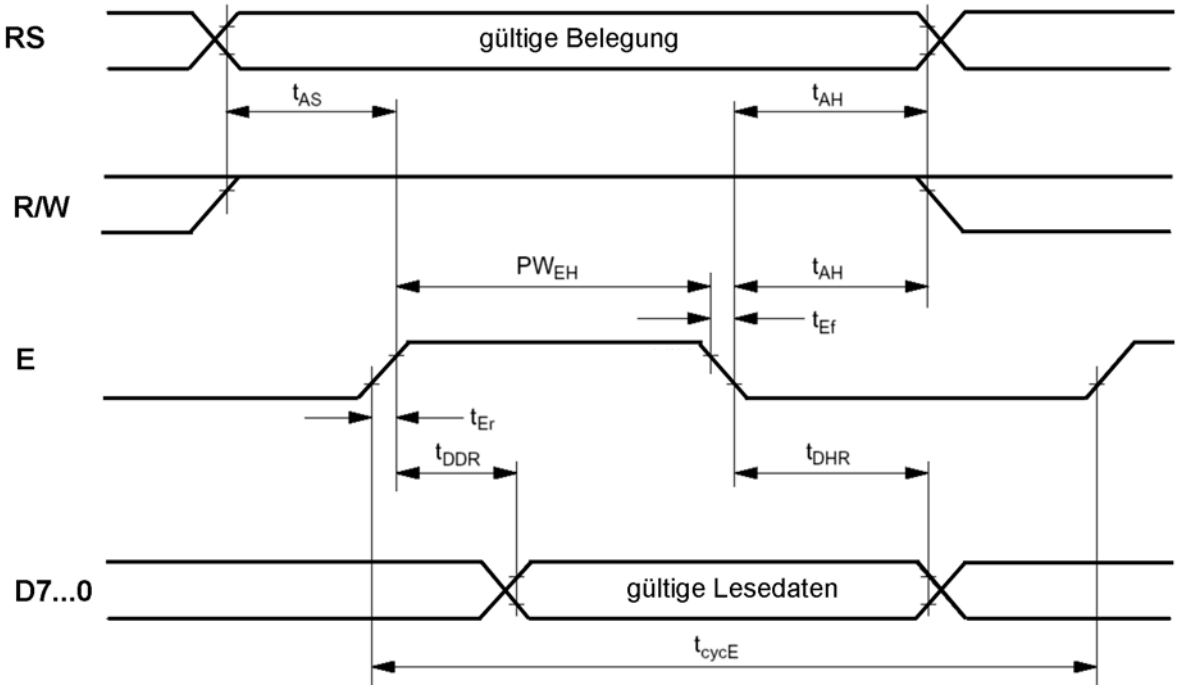

Abb. 8 Der Ablauf eines Lesezugriffs

Zur Programmierpraxis
Es gibt keine Höchstwerte. Die Signalspiele dürfen beliebig langsam ablaufen. Achtung bei schnellen Controllern – u. U. sind Verzögerungen einzubauen, um die Mindestwerte der Vorhaltezeit, Impulsbreite, Zykluszeit usw. einzuhalten.

Die Tabellen 1 und 2 enthalten zwei Angaben:
- für den ursprünglichen Controllerschaltkreis HD44780 und entsprechend kompatible Typen,
- für den verbesserten Typ HD44780U.

Im Zweifelsfall auf Nummer Sicher gehen und für den langsameren Typ programmieren (Abb. 9, 10).

Praxistip:
Es kommt auch auf die Anstiegs- und Abfallzeit des ENABLE-Signals (E) an. Richtwert: höchstens 20 ns zwischen 0 V und 0,7 V_{CC}. Ggf. nachmessen (Oszilloskop). Abhilfen: (1) Pull-up-Widerstand, (2) stärkerer Treiber.

Kennwert	Symbol	HD44780 o. kompatibel			HD44780U		
		min.	typ.	max.	min.	typ.	max.
ENABLE-Zykluszeit	t_{cycE}	1000	-	-	500	-	-
ENABLE-Impulsdauer	PW_{EH}	450	-	-	230	-	-
ENABLE-Anstiegs- und Abfallzeit	t_{Er}, t_{Ef}	-	-	25	-	-	20
Adressen-Vorhaltezeit	t_{AS}	140	-	-	40	-	-
Adressen-Haltezeit	t_{AH}	10	-	-	10	-	-
Daten-Verzögerungszeit (Zugriffszeit)	t_{DDR}	-	-	320	-	-	160
Daten-Haltezeit	t_{DHR}	20	-	-	5	-	-

Tabelle 2 Zeitkennwerte des Lesezugriffs. Alle Zeitangaben in ns

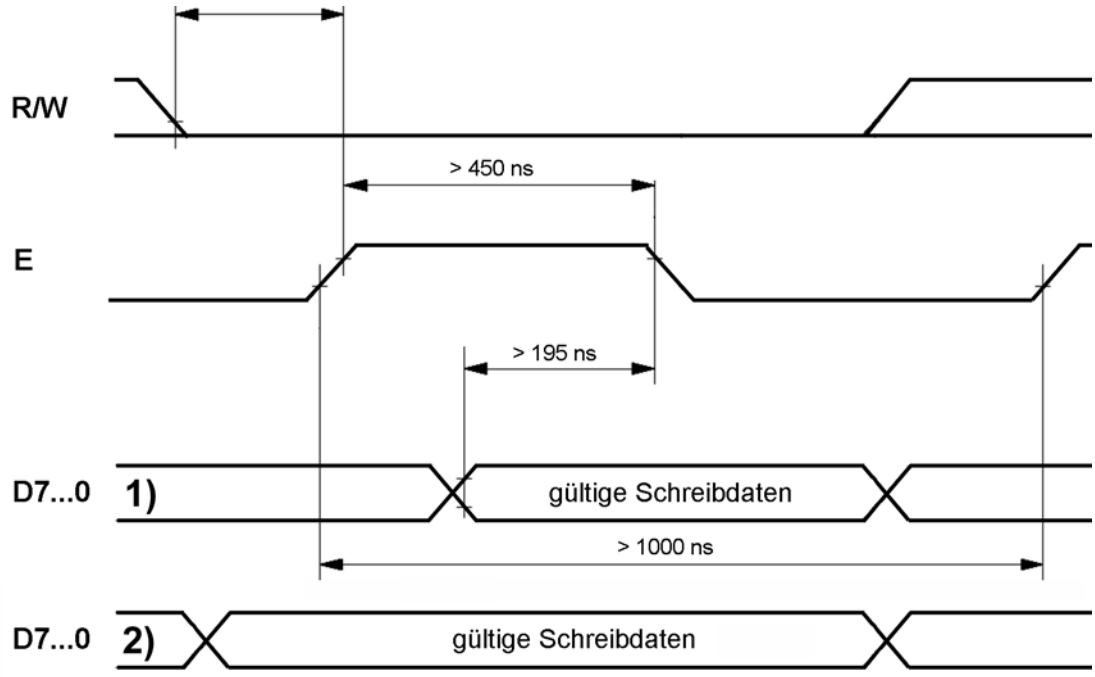

Abb. 9 Der Schreibzugriff aus Sicht des Praktikers. 1) die Schreibdaten können auch bei aktivem E-Impuls bereitgestellt werden (spätestens 195 ns vor dem Abfall des E-Impulses); 2) in der Praxis wird man meist erst die Schreibdaten auf den Bus legen und dann den E-Impuls auslösen

Der typische Ablauf eines Schreibzugriffs:

1. RS je nach Zugriff einstellen. R/W auf 0.
2. Schreibdaten auf den Bus legen.
3. von Schritt 1 an müssen wenigstens 140 ns vergangen sein. E-Impuls erzeugen (mindestens 450 ns).
4. Ggf. Schreibdaten vom Bus nehmen und Bus freigeben.
5. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.

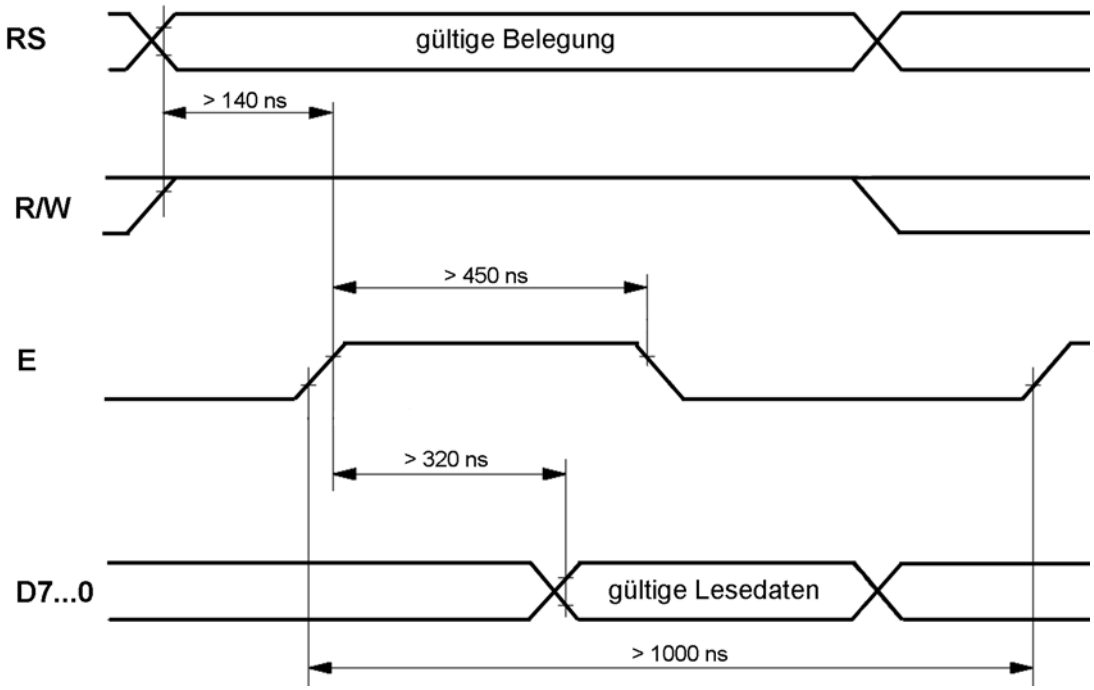

Abb. 10 Der Lesezugriff aus Sicht des Praktikers

Der typische Ablauf eines Lesezugriffs:

1. ggf. Bus freigeben. Betreffenden Port auf Eingang schalten.
2. RS je nach Zugriff einstellen. R/W auf 1.
3. von Schritt 2 an müssen wenigstens 140 ns vergangen sein. E-Signal auf 1.
4. wenigstens 320 ns abwarten.
5. Lesedaten vom Bus abholen.
6. E-Signal auf 0.
7. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.

Praxistip:
Die Zugriffe mit Unterprogrammen (Funktionsaufrufen) erledigen. Ggf. Testprogramme schreiben, die diese Unterprogramme jeweils zyklisch aufrufen (schnellste Aufruffolge). Zeiten mittels Oszilloskop kontrollieren.

1.3 Speicherausstattung

- Datenspeicher (Display Data (DD) RAM): 80 Bytes. 1 Byte = 1 Zeichen.
- fester Zeichengenerator (Character Generator ROM): 9920 Bits bzw. 240 Zeichen, davon:
  - 208 Zeichen im Raster 5 • 8,
  - 32 Zeichen im Raster 5 • 10 (= mit Unterlänge).
- ladbarer Zeichengenerator (Character Generator (CG) RAM): 64 Bytes. Kann aufnehmen:
  - 8 Zeichen im Raster 5 • 8,
    oder
  - 4 Zeichen im Raster 5 • 10 (= mit Unterlänge).

Datenspeicheradressierung
Die Adressierung hängt mit der Zeilenzahl der Anzeige zusammen (Tabelle 3, Abb. 11):

- 1 Zeile: es gibt zwei Ausführungen:
  - fortlaufende Adressierung von 00H...4FH,
  - Aufteilung in zwei Hälften (z. B. 8 + 8 oder 10 + 10 Zeichen). Linke Hälfte beginnt an Adresse 00H, rechte Hälfte an Adresse 40H (Adreßbereiche wie bei zweizeiligen Anzeigen).
- 2 Zeilen: es gibt zwei Adreßbereiche zu 40 Zeichen:
  - 00H...27H,
  - 40H...67H.
- 4 Zeilen: die Adreßbereiche der zweizeiligen Darstellung werden beibehalten. Je Adreßbereich werden zwei Zeilen unterstützt:
  - erste und dritte Zeile: 00H...27H,
  - zweite und vierte Zeile: 40H...67H.

Achtung: Das muß nicht immer stimmen. Datenblattstudium!

Hinweis:
Nicht zur Anzeige ausgenutzte Bytes im Datenspeicher dürfen beliebig belegt werden.

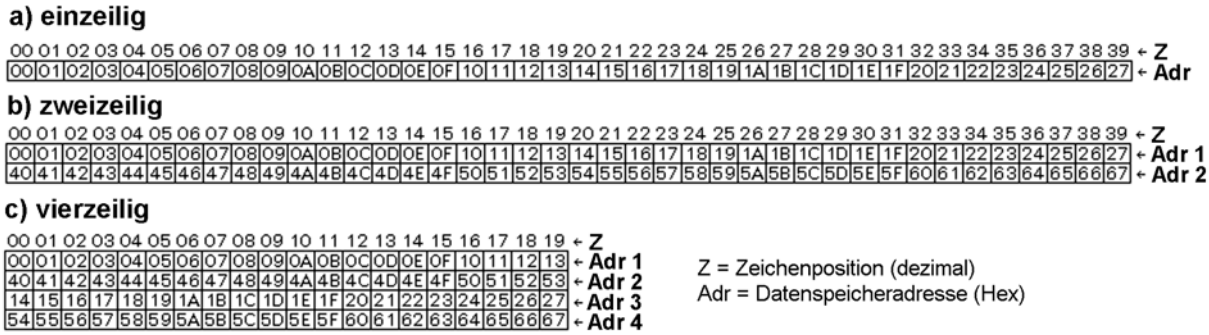

Abb. 11 Datenspeicheradressierung typischer Dotmatrixanzeigen

Noch größere Anzeigeeinheiten (z. B. 4 • 40)
Sie enthalten zwei Controller. Jeder hat ein eigenes ENABLE-Signal (Abb. 12, Tabelle 4). Der erste steuert die 1. und die 2. Zeile an, der zweite die 3. und die 4. Zeile.

Anzeige	1. Zeile	2. Zeile	3. Zeile	4. Zeile
1 • 8	00H...07H			
1 • 16	00H...0FH			
1 • 16 (8 + 8)	00H...07H, 40H...47H			
1 • 20	00H...13H			
1 • 40	00H...27H			
2 • 8	00H...07H	40H...47H		
2 • 12	00H...0BH	40H...4BH		
2 • 16	00H...0FH	40H...4FH		
2 • 20	00H...13H	40H...53H		
2 • 24	00H...17H	40H...57H		
2 • 40	00H...27H	40H...67H		
4 • 16	00H...0FH	40H...4FH	10H...1FH	50H...5FH
4 • 20	00H...13H	40H...53H	14H...27H	54H...67H

Tabelle 3 Datenspeicheradressierung typischer Dotmatrixanzeigen (nach Electronic Assembly)

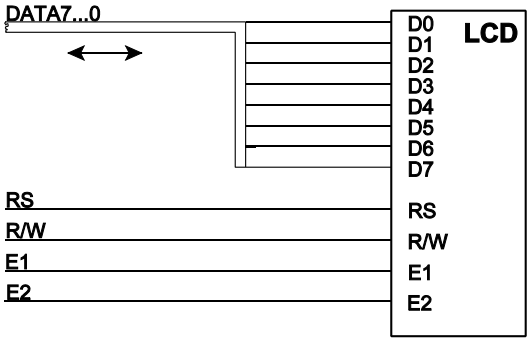

Abb. 12 Das Interface einer größeren LCD-Dotmatrixanzeige (z. B. 4 Zeilen zu 40 Zeichen)

Ansteuerung über	1. Zeile	2. Zeile	3. Zeile	4. Zeile
E1	00H...27H	40H...67H		
E2			00H...27H	40H...67H

Tabelle 4 Datenspeicheradressierung einer 4 • 40-Dotmatrixanzeige (nach Electronic Assembly)

Der feste Zeichenvorrat

Die Schaltkreise werden mit verschiedenen Zeichensätzen gefertigt, auch mit kundenspezifischen (Frage der Stückzahl). Die Abb. 13 und 14 zeigen zwei typische Industriestandard-Zeichensätze. Die Codes der lateinischen Klein- und Großbuchstaben sowie einiger gebräuchlicher Sonderzeichen entsprechen hier dem ASCII-Code (Abb. 15).

Ladbare Zeichen

Es können 8 Zeichen im Raster 5 • 8 oder 4 Zeichen im Raster 5 • 10 geladen werden (Abb. 16, 17).

Abb. 13 Der Zeichensatz A00. Lateinisch + Umlaute + Griechisch + Katakana (nach Hitachi). Die Codes 20H...7DH entsprechen ASCII. Die Codes E0...FF enthalten Zeichen mit Unterlänge (Raster 5 • 10)

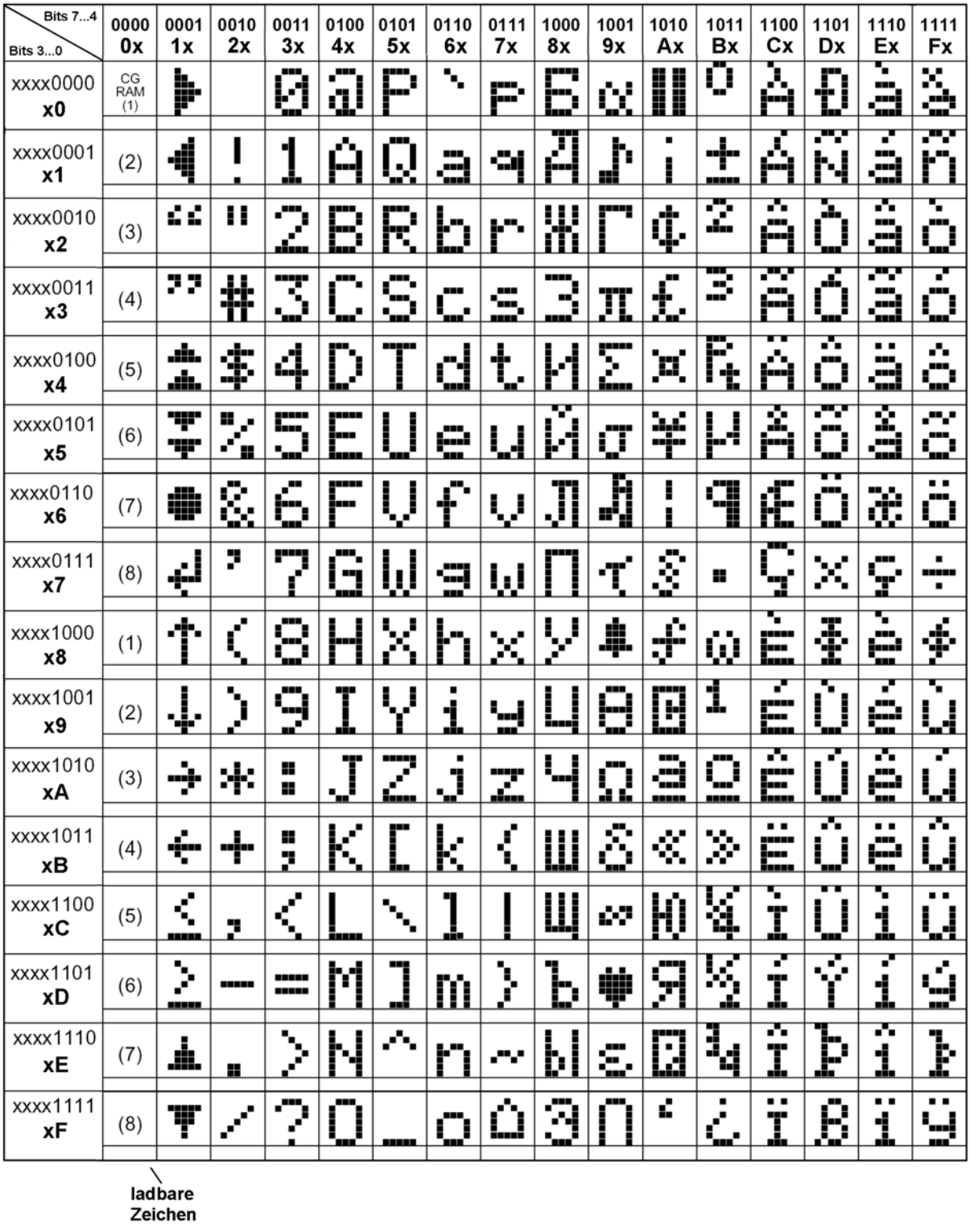

Abb. 14 Der Zeichensatz A02. Lateinisch + Umlaute + Griechisch + Kyrillisch (nach Hitachi). Zeichen 20H...7EH entsprechen ASCII

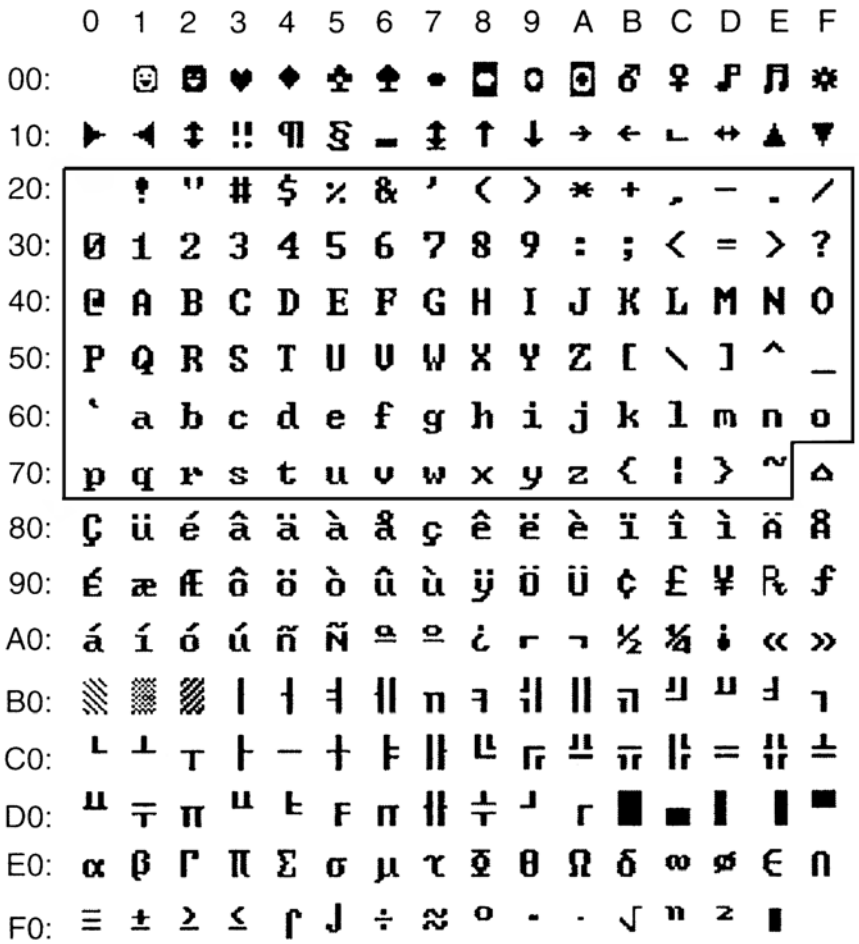

Abb. 15 Zum Vergleich: der erweiterte IBM-Zeichensatz. Eingerahmt: der ursprüngliche Zeichenvorrat nach ASCII

ASCII = American Standard Code for Information Interchange. Der ursprüngliche ASCII-Code ist an sich ein 7-Bit-Code (Wertebereich 0...127 bzw. 00H ...7FH). In Computern entspricht aber ein Zeichen einem Byte; dabei ist das höchstwertige Bit stets Null. Es ist lediglich der Wertebereich von 20H bis 7EH mit darstellbaren Zeichen belegt (maximal 95 verschiedene Zeichen). Mit dem PC hat IBM einen erweiterten Zeichensatz eingeführt, in dem die verbleibenden Belegungen ausgenutzt werden, zum zusätzliche Zeichen zu codieren. Die Spalte entspricht dem niederwertigen, die Zeile dem höherwertigen Halbbyte. Beispiel: Zeichen “d”: Zeile = 60H + Spalte = 4H = 64H. 20H ist der Code des Leerzeichens (Space). Die Codes 20H...7EH (durch Einrahmung gekennzeichnet) entsprechen dem ursprünglichen ASCII-Code.

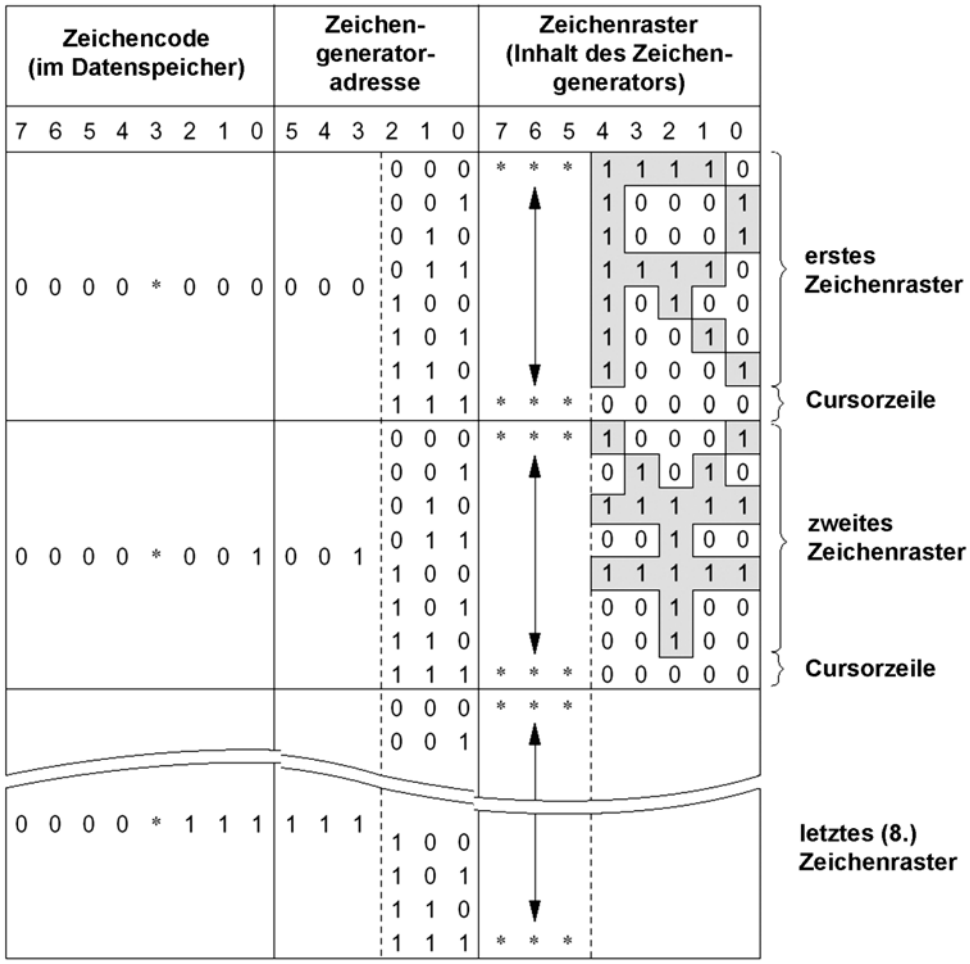

Abb. 16 Ladbare Zeichen im Raster 5 • 8 (nach Hitachi)

Zeichencodes:
1. Zeichen: 00H, 2. Zeichen: 01H usw. 8. Zeichen: 07H. Bit 3 des Zeichencodes ist bedeutungslos (don‘t care). Hierdurch ergibt sich je Zeichen ein weiterer zulässiger Code (für das 1. Zeichen 08H, für das zweite Zeichen 09H usw.).

Laden des Zeichengenerators:
Byteweise. Jedes Byte entspricht einer Zeile des Zeichenrasters. Das Raster belegt nur die Bits 4...0. Belegung: 0 = kein Pixel, 1 = Pixel. Die Bits 7...5 sind bedeutungslos (don‘t care). Die achte Rasterzeile ist die Cursorzeile. Sie wird ggf. vom Cursor überlagert. Typisches Zeichenraster ohne Cursor: 5 • 7.

Zeichengeneratoradressierung:
1. Zeichen: Adressen 0...7, 2. Zeichen: Adressen 8...15 usw.

Adresse der ersten Rasterzeile des n-ten Zeichens (n = 1...8): (n-1) • 8.

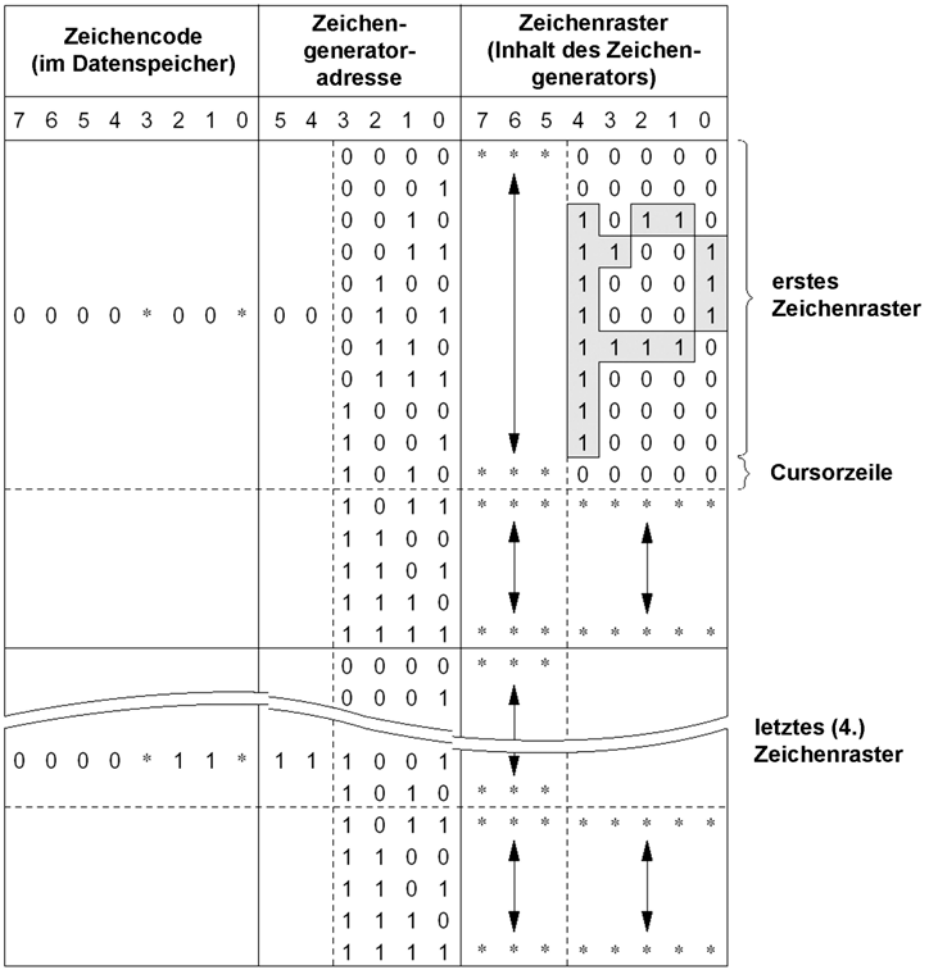

Abb. 17 Ladbare Zeichen im Raster 5 • 10 (nach Hitachi)

Zeichencodes:
1. Zeichen: 00H, 2. Zeichen: 02H; 3. Zeichen: 04H; 4. Zeichen: 06H. Die Bits 3 und 0 des Zeichencodes sind bedeutungslos (don‘t care). Hierdurch ergeben sich je Zeichen drei weitere zulässige Codes, z. B. für das 1. Zeichen 01H, 08H und 09H.

Laden des Zeichengenerators:
Byteweise. Jedes Byte entspricht einer Zeile des Zeichenrasters. Das Raster belegt nur die Bits 4...0. Belegung: 0 = kein Pixel, 1 = Pixel. Die Bits 7...5 sind bedeutungslos (don‘t care). Die zehnte Rasterzeile ist die Cursorzeile. Sie wird ggf. vom Cursor überlagert. Typisches Zeichenraster ohne Cursor: 5 • 9.

Zeichengeneratoradressierung:
Jedes 5 • 10-Zeichen belegt den Platz von zwei 5 • 8-Zeichen, also insgesamt 16 Rasterzeilen = 16 Bytes 1. Zeichen: Adressen 0...15, 2. Zeichen: Adressen 16...31 usw. Die elfte bis 16. Rasterzeile wird nicht dargestellt, deshalb ist der Zeichengeneratorinhalt an den entsprechenden Adressen bedeutungslos.

Adresse der ersten Rasterzeile des n-ten Zeichens (n = 1...4): (n-1) • 16.

1.4 Kommandos

Kommando	Steuerl.		Datenbyte								Beschreibung	max. Ausführungszeit[2)]
	RS	R/W	7	6	5	4	3	2	1	0		
Clear Display	0	0	0	0	0	0	0	0	0	1	gesamte Anzeige löschen. Datenspeicheradresse auf Null	1,52 / 1,64 ms
Return Home	0	0	0	0	0	0	0	0	1	*[1)]	Datenspeicheradresse auf Null. Anzeige an Originalposition (keine Verschiebung). Datenspeicherinhalt bleibt erhalten	1,52 / 1,64 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Einstellung der Bewegungsrichtung des Cursors (I/D). Wahl zwischen Cursorbewegung und Verschieben der Anzeige (S)	37 / 40 µs
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Anzeige ein/aus (D), Cursor ein/aus (C), Blinken an Cursorposition ein/aus (B)	37 / 40 µs
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*[1)]	*[1)]	Cursorbewegung und Verschieben der Anzeige	37 / 40 µs
Function Set	0	0	0	0	1	DL	N	F	*[1)]	*[1)]	Einstellung der Zugriffsbreite (DL), der Zeilenzahl (N) und des Zeichensatzes (F)	37 / 40 µs
CG RAM Adrs Set	0	0	0	1	Zeichengeneratoradresse						Adresse des ladbaren Zeichengenerators einstellen	37 / 40 µs
DD RAM Adrs Set	0	0	1	Datenspeicheradresse							Datenspeicheradresse einstellen	37 / 40 µs
Busy Flag / Adrs Read	0	1	BF	Adreßzähler							Busy-Bit (BF) und aktuelle Adreßzählerbelegung lesen	-
CG RAM / DD RAM Data Write	1	0	zu schreibendes Byte								Schreiben in ausgewählten Speicher	37 / 40 µs
CG RAM / DD RAM Data Write	1	1	gelesenes Byte								Lesen des ausgewählten Speichers	37 / 40 µs

1): bedeutungslos (don‘t care); 2): 44780U / herkömmliche Ausführungen

Tabelle 5 Kommandoübersicht

Clear Display

RS	R/W	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1

Der gesamte Datenspeicher wird mit Leerzeichen (20H) gefüllt. Die Cursoradresse wird auf Null gestellt. Anzeige in Originalposition (keine Verschiebung). Cursor erscheint in der linken (oberen) Ecke. Cursorbewegung (I/D) wird auf Vorwärtsrichtung gestellt (Increment).

Return Home

RS	R/W	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	*

Die Cursoradresse wird auf Null gestellt. Anzeige in Originialposition (keine Verschiebung). Cursor erschein in der linken (oberen) Ecke. Datenspeicherinhalt bleibt erhalten.

Entry Mode Set

RS	R/W	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	I/D	S

Einstellen der Betriebsart bei Zeicheneingabe:
- I/D = 0: Cursoradresse zählt abwärts (Autodecrement) bzw. Rechtsverschiebung,
- I/D = 1: Cursoradresse zählt aufwärts (Autoincrement) bzw. Linksverschiebung
- S = 0: kein Verschieben der Anzeige.
- S = 1: Anzeige wird verschoben (Abb. 18). Bit I/D bestimmt die Schieberichtung.

Ist S = 0, bleibt die Anzeige stehen, und der Cursor wandert, ist S = 1, bleibt der Cursor stehen und die Anzeige wandert.

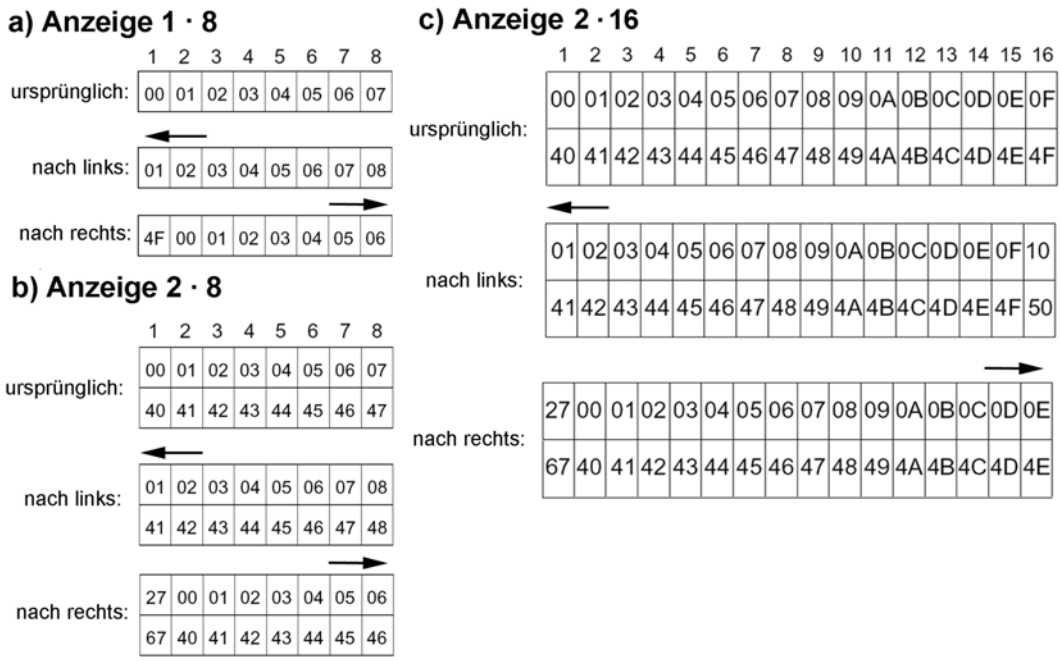

Abb. 18 Das Verschieben der Anzeige anhand von Beispielen (nach Hitachi)

Hinweise:

1. Der Speicherinhalt verbleibt auf seinen ursprünglichen Adressen; das Verschieben geschieht durch entsprechend versetztes Adressieren beim Auslesen zur Anzeige.
2. Alle Zeilen werden gleichermaßen verschoben.

3. Das Verschieben ist praktisch ein Rotieren über den gesamten Speicherbereich (von Adresse 0 zur Endadresse und wieder zu Adresse 0 (Wrap Around).
4. Verschiebebereich bei einzeiliger Darstellung: 0...4F (80 Zeichen);
5. Verschiebebereiche bei zweizeiliger Darstellung: oben: 0...27H , unten: 40H...67H (zweimal 40 Zeichen).

Display On/Off Control

RS	R/W
0	0

7	6	5	4	3	2	1	0
0	0	0	0	1	D	C	B

Ein- und Ausschalten der Anzeige, Steuerung der Cursordarstellung:
- D = 0: Anzeige ausgeschaltet. Daten bleiben erhalten.
- D = 1: Anzeige eingeschaltet.
- C = 0: Cursor wird nicht angezeigt.
- C = 1: Cursor wird angezeigt (Abb. 19a, b).
- B = 0: stehende Cursordarstellung.
- B = 1: blinkende Zeichendarstellung an Cursorposition (Abb. 19c).

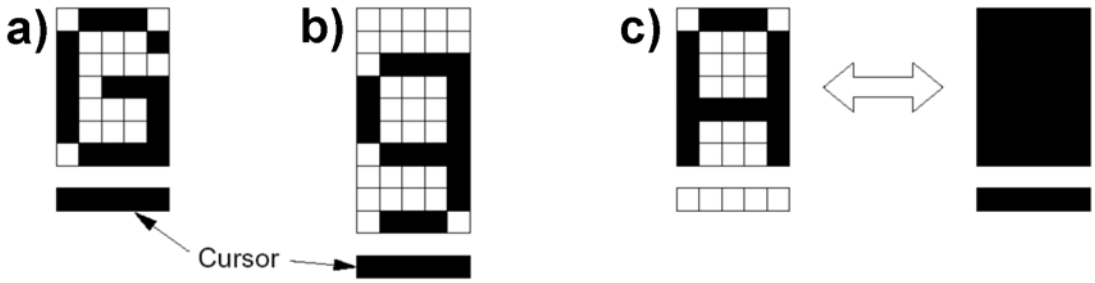

Abb. 19 Cursordarstellung (nach Hitachi). a) Zeichenraster 5 • 8; b) Zeichenraster 5 • 10; c) blinkende Darstellung

Cursor/Display Shift

RS	R/W
0	0

7	6	5	4	3	2	1	0
0	0	0	1	S/C	R/L	*	*

Bewegen des Cursors oder der gesamten Anzeige:
- S/C = 0: Cursor bewegen. Anzeige bleibt stehen. Wenn R/L = 0, wird Cursoradresse vermindert, wenn R/L = 1, wird Cursoradresse erhöht.
- S/C = 1: Anzeige bewegen. Cursoradresse bleibt stehen, Cursor wandert mit der Anzeige.
- R/L = 0: Bewegen nach links,
- R/L = 1: Bewegen nach rechts.

Function Set

RS	R/W
0	0

7	6	5	4	3	2	1	0
0	0	1	DL	N	F	*	*

Betriebsarteneinstellung:
- DL = 0: Zugriffsbreite = 4 Bits (7...4).
- DL = 1: Zugriffsbreite = 8 Bits (7...0).
- N = 0: einzeilige Anzeige = fortlaufende Datenspeicheradressierung (mit 00H beginnend). Für einzeilige Anzeigen.
- N = 1: zweizeilige Anzeige = geteilte Datenspeicheradressierung (mit 00H und 40H beginnend). Für zwei- und vierzeilige Anzeigen sowie für einzeilige Typen mit Organisation 8 + 8 .
- F = 0: Zeichenraster 5 • 8,
- F = 1: Zeichenraster 5 • 10. Nur einzeilige Anzeige möglich (N ist bedeutungslos).

CG RAM Adrs Set

RS	R/W
0	0

7	6	5	4	3	2	1	0
0	1	Zeichengeneratoradresse					

Die Zeichengeneratoradresse wird gemäß der Belegung der Datenbits 5...0 eingestellt. Nachfolgende Datenzugriffe (Schreiben/Lesen) beziehen sich auf den ladbaren Zeichengenerator.

DD RAM Adrs Set

RS	R/W
0	0

7	6	5	4	3	2	1	0
1	Datenspeicheradresse						

Die Datenspeicheradresse wird gemäß der Belegung der Datenbits 6...0 eingestellt.. Nachfolgende Datenzugriffe (Schreiben/Lesen) beziehen sich auf den Datenspeicher. Die eingestellte Datenspeicheradresse wird zur Cursoradresse.

Der Adreßzähler
Für Zeichengenerator- und Datenspeicherzugriffe wird ein gemeinsamer Adreßzähler (Address Counter AC) verwendet. Der Inhalt dieses Zählers bestimmt zugleich die Cursorposition. (Wird zwischendurch der Zeichengenerator umgeladen, anschließend ggf. die Cursorposition erneut einstellen.)

Busy Flag / Adrs Read

RS	R/W
0	1

7	6	5	4	3	2	1	0
BF	Adreßzähler (AC)						

Lesen der Besetztanzeige (Busy Flag BF) und der aktuellen Adreßzählerbelegung.
- BF = 0: Anzeige ist bereit, ein neues Kommando entgegenzunehmen.
- BF = 1: Anzeige ist mit der Ausführung eines Kommandos beschäftigt. Kann nur Kommando Busy Flag / Adrs Read ausführen. Kein anderes Kommando übertragen!

CG RAM / DD RAM Data Write

RS	R/W
1	0

7	6	5	4	3	2	1	0
Schreibdaten							

Schreiben eines Bytes. Welcher Speicher damit geladen wird, hängt von der letzten Adreßübertragung ab (Kommandos CG RAM Adrs Set oder DD RAM Adrs Set). Nach dem Schreiben wird der Adreßzählerinhalt automatisch erhöht (+ 1) oder vermindert (-1). Steuerung über Bit I/D. Beim Schreiben in den Datenspeicher wird die Cursorposition geändert oder die Anzeige verschoben (Steuerung über Bit S). Stellen der Bits I/D und S: mittels Kommando Entry Mode Set.

CG RAM / DD RAM Data Read

RS	R/W
1	1

7	6	5	4	3	2	1	0
Lesedaten							

Lesen eines Bytes. Welcher Speicher gelesen wird, hängt von der letzten Adreßübertragung ab (Kommandos CG RAM Adrs Set oder DD RAM Adrs Set). Nach dem Lesen wird der Adreßzählerinhalt automatisch erhöht (+ 1) oder vermindert (-1). Steuerung über Bit I/D (Kommando Entry Mode Set). Die Anzeige wird nicht verschoben (auch nicht bei gesetztem S-Bit).

Programmierhinweis:
Vor dem ersten Lesezugriff ein Kommando CG RAM Adrs Set oder DD RAM Adrs Set geben, um die Anfangsadresse einzutragen.

1.5 Rücksetzen beim Einschalten

Der Schaltkreis gelangt in folgenden Anfangzustand:

- Datenspeicher gelöscht (Inhalt H).
- DL = 1: Zugriffsbreite = 8 Bits.
- N =0: einzeilige Anzeige.
- F = 0: Zeichenraster 5 • 8.
- D = 0: Anzeige aus.
- C = 0: Cursor aus.
- B = 0: Blinken aus.
- I/D = 1: Adreßerhöhung (Aufwärtszählung).
- S = 0: kein Schieben der Anzeige.

Hinweis:
Das Rücksetzen wird nur dann wirksam, wenn die Speisespannung schnell genug hochläuft. Richtwert: 0,1 ms ... 10 ms für den Anstieg von 0,2 V auf 0,9 V_{CC}. Sicherheitshalber stets programmseitig initialisieren (Abschnitt 1.7).

1.6 Zugriffsabläufe

Die Zeichendarstellung wird durch Folgen entsprechender Kommandos aufgebaut. Nach dem Senden eines Kommandos ist die Anzeige zunächst mit dessen Ausführung beschäftigt und kann kein weiteres Kommando annehmen (Besetztzustand). Es gibt zwei Möglichkeiten, Kommandofolgen zu programmieren (Abb. 20):

a) Warten:
- das erste Kommando übertragen,
- lange genug warten (gemäß maximaler Ausführungszeit (Tabelle 5)),
- das zweite Kommando übertragen,
- lange genug warten usw.

b) das Busy-Bit (BF) abfragen:
- das erste Kommando übertragen
- das Busy-Bit solange abfragen, bis es = 0 ist,
- das zweite Kommando übertragen,
- das Busy-Bit solange abfragen, bis es = 0 ist usw.

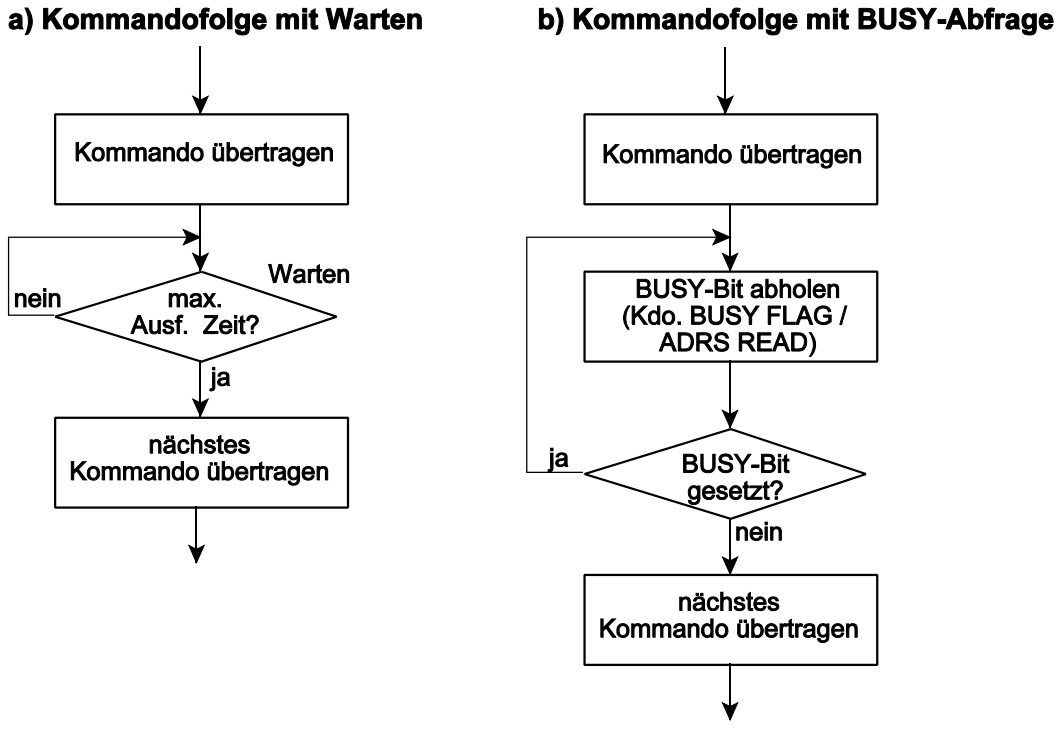

Abb. 20 Übertragung von Kommandofolgen

Zugriffsbreite = 8 Bits
Jede Kommandoübertragung und jede Busy-Abfrage ist ein einziger Zugriff (Abb. 21; vgl. auch die Abb. 7 bis 10).

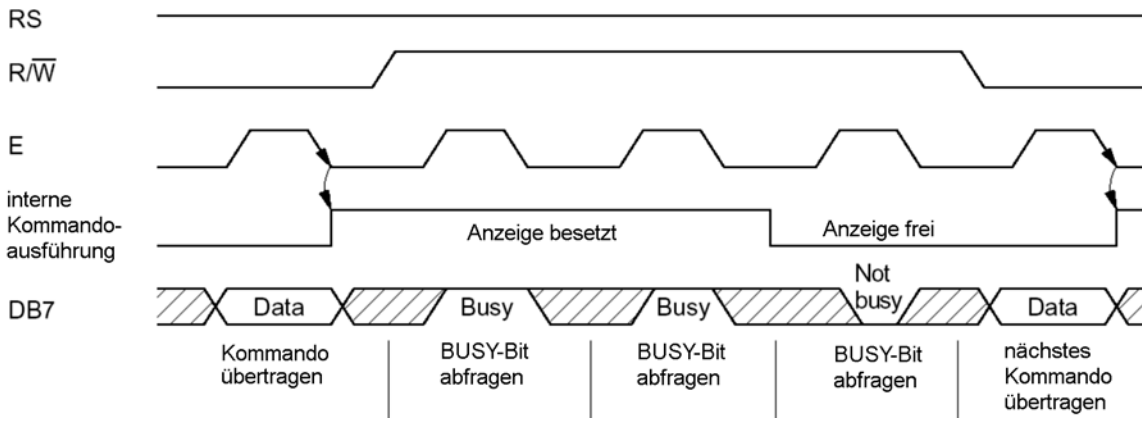

Abb. 21 Kommandoausführung bei 8 Bits Zugriffsbreite (nach Hitachi)

Hinweis:
Jede Busy-Abfrage muß ein vollständiger Lesezugriff (mit E-Impuls) sein.

Zugriffsbreite = 4 Bits
Jede Kommandoübertragung und jede Busy-Abfrage erfordert zwei Zugriffe, die unmittelbar aneinandergehängt werden können. Zuerst werden die Datenbits 7...4 übertragen, dann die Datenbits 3...0 (Abb. 22 bis 24).

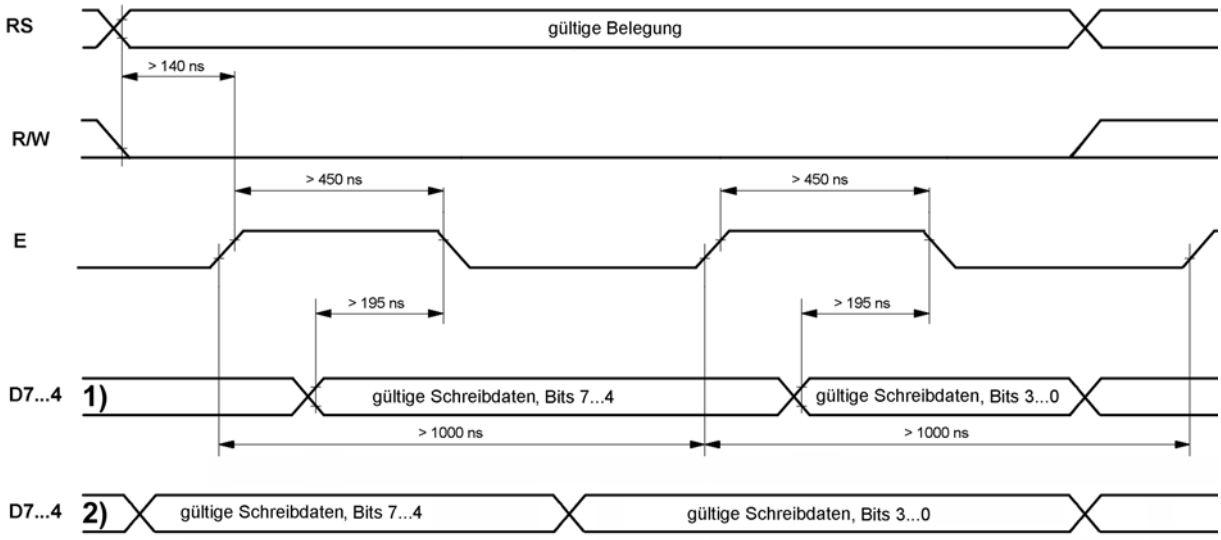

Abb. 22 Schreibzugriff bei 4 Bits Zugriffsbreite (Zeitraster gemäß Abb. 9). 1) die Schreibdaten können auch bei aktivem E-Impuls bereitgestellt werden (spätestens 195 ns vor dem Abfall des E-Impulses); 2) in der Praxis wird man meist erst die Schreibdaten auf den Bus legen und dann den E-Impuls auslösen

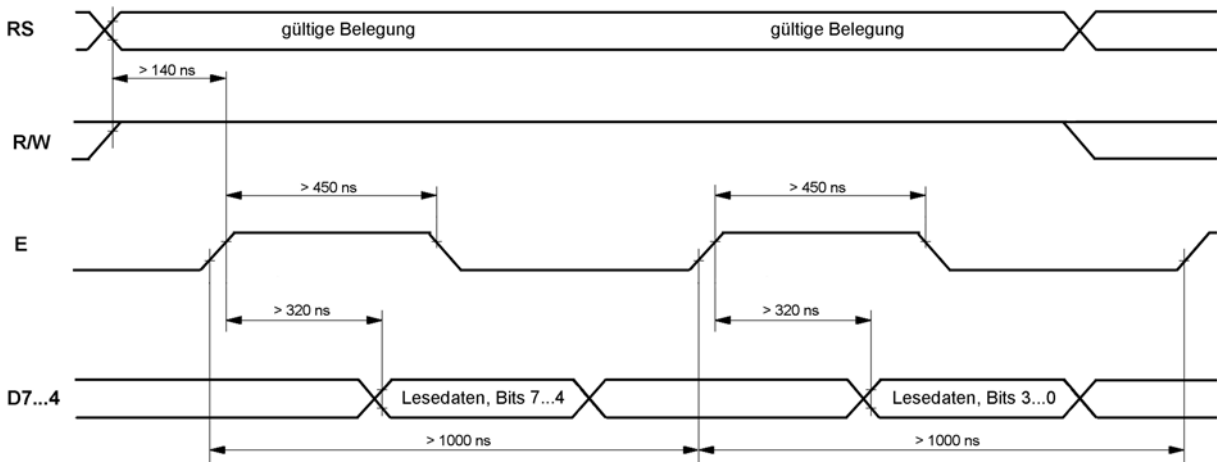

Abb. 23 Lesezugriff bei 4 Bits Zugriffsbreite (Zeitraster gemäß Abb. 10)

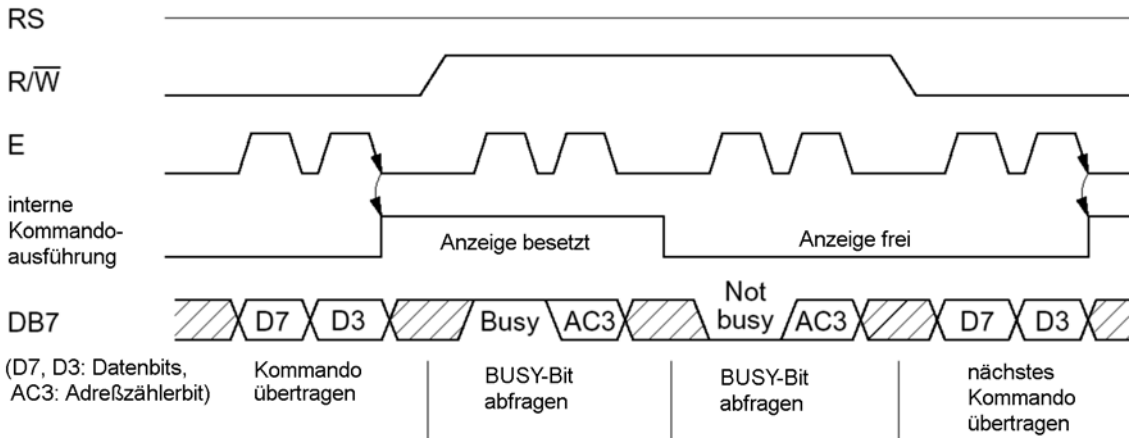

Abb. 24 Kommandoausführung bei 4 Bits Zugriffsbreite (nach Hitachi)

Hinweis:
Jede Busy-Abfrage muß ein vollständiger Lesezugriff (mit zwei E-Impulsen) sein. Das Busy-Bit erscheint während des ersten E-Impulses auf Datenbusleitung DB7.

1.7 Initialisierung

Nach dem Einschalten der Speisespannung wird die Anzeige intern zurückgesetzt (vgl. Abschnitt 1.5). Sollte das nicht funktionieren (z. B. infolge einer zu langsam ansteigenden Speisespannung), empfiehlt Hitachi ein programmseitigen Rücksetzablauf (Abb. 25).

Die eigentliche Initialisierung besteht darin, die jeweils gewünschte Betriebsart einzustellen (Tabellen 7 und 8).

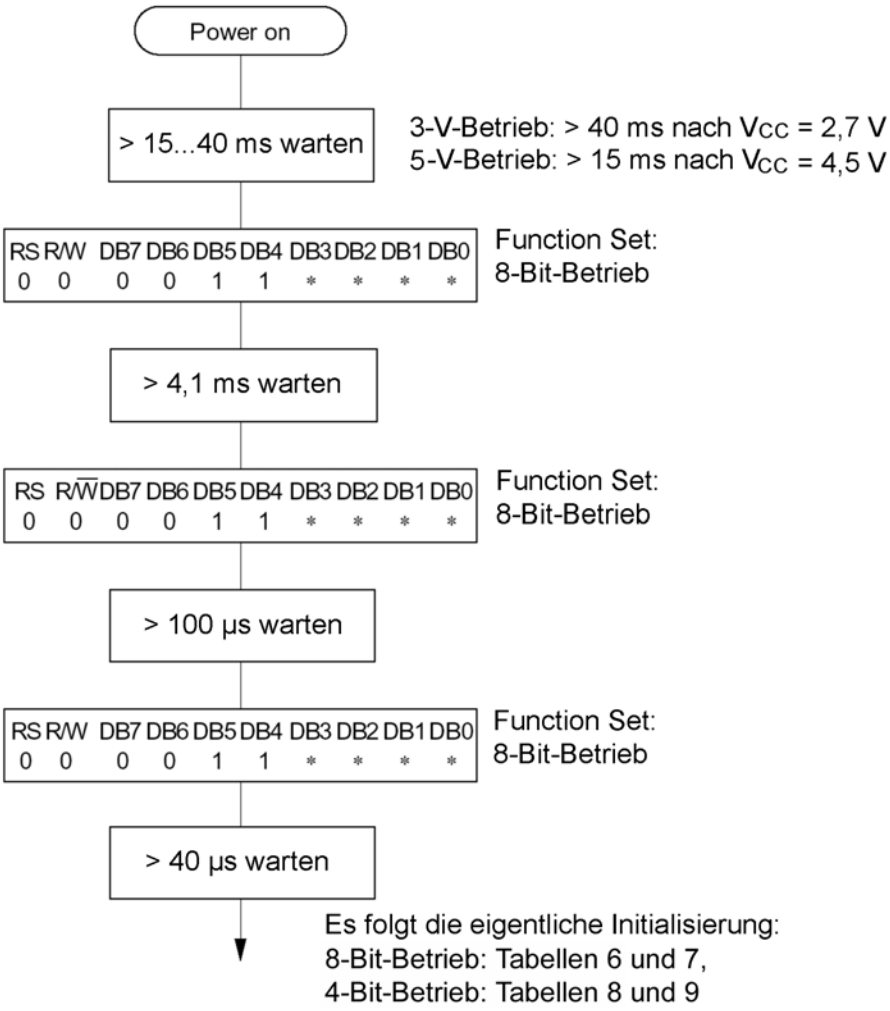

Abb. 25 Programmseitiges Rücksetzen nach dem Einschalten (nach Hitachi). Wird dann empfohlen, wenn das interne Rücksetzen nicht richtig funktioniert (z. B. weil die Speisespannung zu langsam hochläuft). Kurz nach dem Einschalten wird die Abfrage des Busy-Bits noch nicht unterstützt

Kommando	Steuerl.		Datenbyte								Anmerkungen
	RS	R/W	7	6	5	4	3	2	1	0	
Function Set	0	0	0	0	1	1	N	F	0	0	8-Bit-Betrieb. Zeilenzahl N und Zeichenraster F nach Bedarf
Clear Display	0	0	0	0	0	0	0	0	0	1	Datenspeicher löschen. Cursor auf erste Zeichenposition (links (oben))
Display On/Off Control	0	0	0	0	0	0	1	1	C	B	Anzeige ein. Cursordarstellung C und Blinken B nach Bedarf
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	I/D und S nach Bedarf

Tabelle 6 Initialisierung für 8-Bit-Betrieb. Nach jedem Kommando Ausführungszeit abwarten oder das Busy-Bit abfragen (vgl. die Abb. 20 und 21)

Kommando	Steuerl.		Datenbyte								Anmerkungen
	RS	R/W	7	6	5	4	3	2	1	0	
Function Set	0	0	0	0	1	1	1	0	0	0	8-Bit-Betrieb. 2 Zeilen, Zeichenraster 5 • 8
Clear Display	0	0	0	0	0	0	0	0	0	1	Datenspeicher löschen. Cursor auf erste Zeichenposition (links (oben))
Display On/Off Control	0	0	0	0	0	0	1	1	1	1	Anzeige ein, Cursordarstellung ein, Cursor blinken
Entry Mode Set	0	0	0	0	0	0	0	1	1	0	Cursoradresse zählt aufwärts (Autoincrement); kein Schieben

Tabelle 7 Initialisierungsbeispiel 8-Bit-Betrieb

Kommando	Steuerl.		Datenbyte				Beschreibung
	RS	R/W	7	6	5	4	
Function Set	0	0	0	0	1	0	noch 8-Bit-Betrieb. 4-Bit-Betrieb wird eingestellt
Function Set	0	0	0	0	1	0	4-Bit-Betrieb. Zeilenzahl N und Zeichensatz F nach Bedarf
			N	F	0	0	
Clear Display	0	0	0	0	0	0	Datenspeicher löschen. Cursor auf erste Zeichenposition (links (oben))
			0	0	0	1	
Display On/Off Control	0	0	0	0	0	0	Anzeige ein. Cursordarstellung C und Blinken B nach Bedarf
			1	1	C	B	
Entry Mode Set	0	0	0	0	0	0	I/D und S nach Bedarf
			0	1	I/D	S	

Tabelle 8 Initialisierung für 4-Bit-Betrieb. Mit Ausnahme des ersten erfordert jedes Kommando zwei Zugriffe (vgl. die Abb. 22 und 23). Nach jedem Kommando Ausführungszeit abwarten oder das Busy-Bit abfragen (vgl. die Abb. 20 und 24)

Achtung:
Wenn sich die Anzeige bereits im 4-Bit-Betrieb befindet, darf das erste Function-Set-Kommando von Tabelle 8 nicht mehr ausgeführt werden. Fallbeispiel: wiederholter Durchlauf des gesamten Anzeigeprogramms bei eingeschalteter Anzeige, z. B. während der Programmentwicklung (Debugging) oder bei erneuter Initialisierung (z. B. Wiederanlauf nach Watchdog-Rücksetzen). Abhilfe: (1) Anzeige vor jedem neuen Programmstart kurzzeitig ausschalten, (2) nach dem ersten Durchlauf das erste Function-Set-Kommado auskommentieren o. dergl., (3) vor Verlassen des Programms die Anzeige wieder in den 8-Bit-Modus schalten (Kommando Function Set mit Datenbyte 3xH), (4) beim Wiederanlauf (z. B. nach dem Watchdog-Rücksetzen) dieses Kommando nicht ausführen..

Kommando	Steuerl.		Datenbyte				Beschreibung
	RS	R/W	7	6	5	4	
Function Set	0	0	0	0	1	0	noch 8-Bit-Betrieb. 4-Bit-Betrieb wird eingestellt
Function Set	0	0	0	0	1	0	4-Bit-Betrieb. 2 Zeilen, Zeichenraster 5 • 8
			1	0	0	0	
Clear Display	0	0	0	0	0	0	Datenspeicher löschen. Cursor auf erste Zeichenposition (links (oben))
			0	0	0	1	
Display On/Off Control	0	0	0	0	0	0	Anzeige ein, Cursordarstellung ein, Cursor blinken
			1	1	1	1	
Entry Mode Set	0	0	0	0	0	0	Cursoradresse zählt aufwärts (Autoincrement); kein Schieben
			0	1	1	0	

Tabelle 9 Initialisierungsbeispiel 4-Bit-Betrieb

1.8 Vermischte Hinweise

Was Dotmatrixanzeigen nicht können (1): Inversdarstellung
Beschränkt die Nutzbarkeit für Bedienkonzepte, die auf Menü-Auswahl beruhen.

Erklärung:
Es ist oftmals erforderlich, die aktuelle Auswahl im Menü hervorzuheben. Typische Varianten:

- Markieren durch vor- und nachgesetzte Zeichen (beispielsweise < und >),
- entsprechendes Positionieren des Cursors,
- Darstellung mit abweichender Helligkeit (heller oder dunkler),
- blinkende Darstellung (der gesamten Anzeige),
- Inversdarstellung. Zumeist die ansehnlichste Darstellungsart.

Abhilfe: Entsprechende "inverse" Zeichen in den ladbaren Zeichengenerator (CG RAM) transportieren. Die jeweilige Darstellung muß allerdings mit einem Zeichensatz aus maximal 8 Zeichen auskommen...

Was Dotmatrixanzeigen nicht können (2): Blinkende Darstellung
Abhilfe: mit Software blinken. Die betreffenden Zeichenpositionen programmseitig zyklisch zwischen dem eigentlichen Inhalt und Leerzeichen (20H) umladen. Blinkfrequenz 3... 10 Hz (ausprobieren).

Ausnutzung des Datenspeichers
Die 80 Bytes sind immer vorhanden, auch wenn weniger Zeichen darstellbar sind. Nutzungsmöglichkeiten:

- als verlängerte Anzeige (Scrollfunktionen). Hierzu kann ggf. der Verschiebebetrieb ausgenutzt werden.
- als allgemeiner Datenpuffer.

Ausnutzung des Verschiebebetriebs
Lohnt sich nur bei kleineren Anzeigen (ein- oder zweizeilig) – und da auch nur dann, wenn ein besonders kleiner Mikrocontroller (ohne ausreichende RAM-Ausstattung) trickreich ausgenutzt werden muß. Bei zweizeiligen Anzeigen ist der praktische Nutzen schon fraglich.

Tip zur Programmorganisation
Im steuernden Controller eine Art Bildspeicher (als Zeichen-Array) vorsehen und wie einen Video-Bildspeicher betreiben. Dort alle Sonderaktionen erledigen. Beispiel: das Scrollen der Darstellung. Im Gegensatz zur eingebauten Verschiebefunktion läßt sich hier jeder beliebige Spezialeffekt programmieren: rauf/runter, rechts/links, Verschieben über Zeilengrenzen usw. Anzeige bedarfsweise durch Blocktransporte aktualisieren.

Richtwerte zum programmseitigen Aktualisieren der Anzeige:

- 1 Byte erfordert rund 50 µs,
- 16 Bytes erfordern 800 µs (ca. 1 ms),
- 80 Bytes erfordern 4...5 ms.

2. Graphische Anzeigen

Viele einfache graphische Anzeigen sind mit Display-Controllern HD61202 (Hitachi) oder kompatiblen Schaltkreisen bestückt. Der einzelne Controller-Schaltkreis unterstützt eine Darstellung aus maximal 64 • 64 = 4096 Bildpunkten. Größere Anzeigen enthalten mehrere Schaltkreise, von denen jeweils einer programmseitig angesprochen werden kann (Schaltkreisauswahl). Das Interface hat einen 8-Bit-Datenbus, 3 Steuersignale, 1 Rücksetzsignal sowie ggf. die erforderliche Anzahl an Schaltkreisauswahlsignalen (Abb. 26). Es gibt keinen 4-Bit-Betrieb.

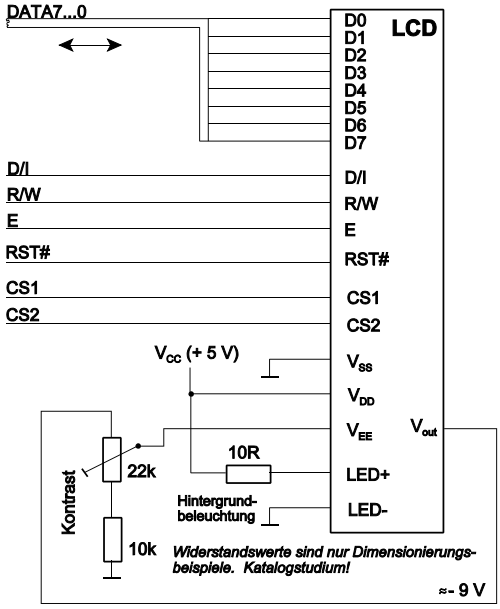

Abb. 26 Graphikanzeige 128 • 64 Pixel mit zwei HD61202

2.1 Interfacesignale im Überblick

DATA7...0
Bidirektionaler Datenbus. Zur Übertragung von Daten, Adressen, Kommandos und Zustandsmeldungen.

Der inaktive Datenbus
Es handelt sich um einen reinen Tri-State-Bus. Der Controllerschaltkreis hat keine Pull-up-Widerstände o. dergl.

Zur Abhilfe:
- externe Pull-up-Widerstände vorsehen,
- Pull-up-Widerstände des Mikrocontroller-Ports anschalten,
- Bus parken (bei Nichtnutzung als Ausgang konfigurieren).

D/I: Data /Instruction
Eingang. Steuert die Art der Übertragung:
- D/I = 0: Kommando- oder Zustandsübertragung,
- D/I = 1: Datenübertragung.

R/W: Read / Write
Eingang zur Lese-Schreib-Steuerung:
- R/W = 0: Schreibzugriff. Bus wird vom Controller nicht aktiviert (kann also mit zu schreibenden Daten belegt werden).
- R/W = 1: Lesezugriff. Wenn E = 1, wird Bus vom Controller aktiviert und mit Lesedaten belegt.

E: Enable
Eingang. Allgemeines Erlaubnissignal:
- E = 0: keine Wirkung. Controller inaktiv (nicht ausgewählt). Datenbus frei.
- E = 1: Wirkung gemäß Belegung von D/I und R/W.

RST#: Reset
Eingang. Direktwirkendes Rücksetzsignal. Minimale Impulsdauer: 1 µs.
- RST# = 0: Rücksetzen. Kein programmseitiger Zugriff möglich. Rücksetzwirkungen:
  - Anzeige aus,
  - Zeilenzähler auf Zeile 0.
- RST# = 1: kein Rücksetzen. Programmseitiger Zugriff möglich.

CS1, CS2: Chip Select
Eingänge. Jeder Eingang wählt einen Controllerschaltkreis aus.
- CS1: Controller für linken Bildhälfte,
- CS2: Controller für rechte Bildhälfte.

Die Signale wirken aktiv high:
- CSx = 0: Schaltkreis nicht ausgewählt,
- CSx = 1: Schaltkreis ausgewählt.

Hinweis:
Der Controllerschaltkreis selbst hat drei Auswahlsignale CS1#, CS2#, CS3, die in konjunktiver Verknüpfung wirken (Schaltkreis ist ausgewählt, wenn Belegung = 0, 0, 1). Je nach Anzeigetyp sind bestimmte Auswahlsignale auf die Anschlüsse geführt. Datenblattstudium! Im folgenden beziehen wir uns auf die Schnittstelle von Abb. 26.

2.2 Signalspiele am Interface

Die Signalspiele entsprechen im wesentlichen denen der Dotmatrixanzeigen. Es gibt im Grunde nur zwei Zugriffsarten: Schreiben und Lesen (Abb. 27 und 28, Tabellen 10 und 11). Alles weitere hängt vom Auswahlsignal D/I und (beim Schreiben) von der Belegung des Datenbytes ab.

Zur Programmierpraxis
Es gibt keine Höchstwerte. Die Signalspiele dürfen beliebig langsam ablaufen. Achtung bei schnellen Controllern – u. U. sind Verzögerungen einzubauen, um die Mindestwerte der Vorhaltezeit, Impulsbreite, Zykluszeit usw. einzuhalten. Vgl. auch die Abb. 9 und 10.

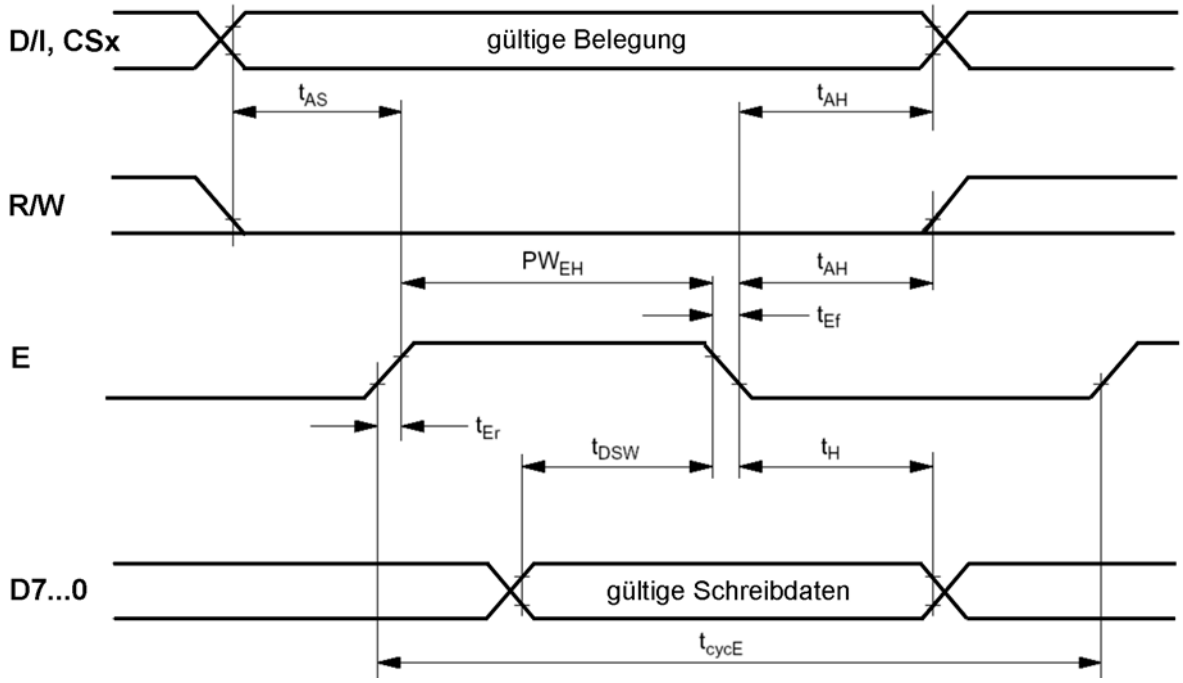

Abb. 27 Der Ablauf eines Schreibzugriffs

Kennwert	Symbol	min.	typ.	max.
ENABLE-Zykluszeit	t_{cycE}	1000	-	-
ENABLE-Impulsdauer	PW_{EH}	450	-	-
ENABLE-Anstieges- und Abfallzeit	t_{Er}, t_{Ef}	-	-	25
Adressen-Vorhaltezeit	t_{AS}	140	-	-
Adressen-Haltezeit	t_{AH}	10	-	-
Daten-Vorhaltezeit	t_{DSW}	200	-	-
Daten-Haltezeit	t_{H}	10	-	-

Tabelle 10 Zeitkennwerte des Schreibzugriffs. Alle Zeitangaben in ns

Der typische Ablauf eines Schreibzugriffs:

1. CS1, CS 2 einstellen (nur jeweils einen Schaltkreis auswählen – Belegung 1, 0 oder 0,1). D/I je nach Zugriff einstellen. R/W auf 0.
2. Schreibdaten auf den Bus legen.
3. von Schritt 1 an müssen wenigstens 140 ns vergangen sein. E-Impuls erzeugen (mindestens 450 ns).
4. Ggf. Schreibdaten vom Bus nehmen und Bus freigeben.
5. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.

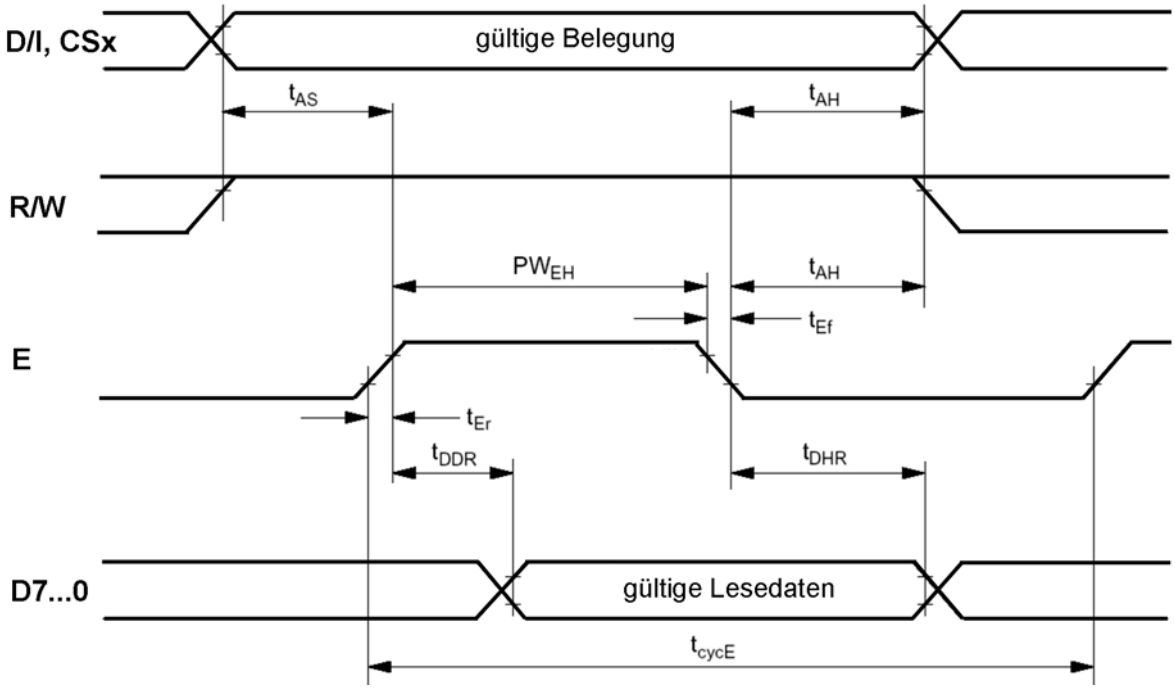

Abb. 28 Der Ablauf eines Lesezugriffs

Kennwert	Symbol	min.	typ.	max.
ENABLE-Zykluszeit	t_{cycE}	1000	-	-
ENABLE-Impulsdauer	PW_{EH}	450	-	-
ENABLE-Anstieges- und Abfallzeit	t_{Er}, t_{Ef}	-	-	25
Adressen-Vorhaltezeit	t_{AS}	140	-	-
Adressen-Haltezeit	t_{AH}	10	-	-
Daten-Verzögerungszeit (Zugriffszeit)	t_{DDR}	-	-	320
Daten-Haltezeit	t_{DHR}	20	-	-

Tabelle 11 Zeitkennwerte des Lesezugriffs. Alle Zeitangaben in ns

Der typische Ablauf eines Lesezugriffs:

1. ggf. Bus freigeben. Betreffenden Port (des Mikrocontrollers) auf Eingang schalten.
2. CS1, CS 2 einstellen (nur jeweils einen Schaltkreis auswählen – Belegung 1, 0 oder 0,1). D/I je nach Zugriff einstellen. R/W auf 1.
3. von Schritt 2 an müssen wenigstens 140 ns vergangen sein. E-Signal auf 1.
4. wenigstens 320 ns abwarten.
5. Lesedaten vom Bus abholen.
6. E-Signal auf 0.
7. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.

2.3 Bildspeicher- und Anzeigeorganisation

Der Bildspeicher ist der einzige programmseitig adressierbare Speicher im Schaltkreis. Jedes Bit entspricht einem Bildpunkt (Pixel).

- Speicherkapazität: 512 Bytes = 512 • 8 Pixel = 4096 Pixel.
- Bildraster: 64 • 64 Pixel.

Bildspeicherorganisation und Bildspeicheradressierung
Der Bildspeicher ist in 8 Seiten zu je 64 Bytes aufgeteilt. Jede Seite entspricht 8 Zeilen zu je 64 Pixeln (Abb. 29).

Die Bildspeicheradresse besteht aus zwei Teilen:

- X-Adresse = Seitenadresse. 3 Bits.
- Y-Adresse = Spaltenadresse (eines vertikalen Blocks aus 8 Pixeln). 6 Bits.

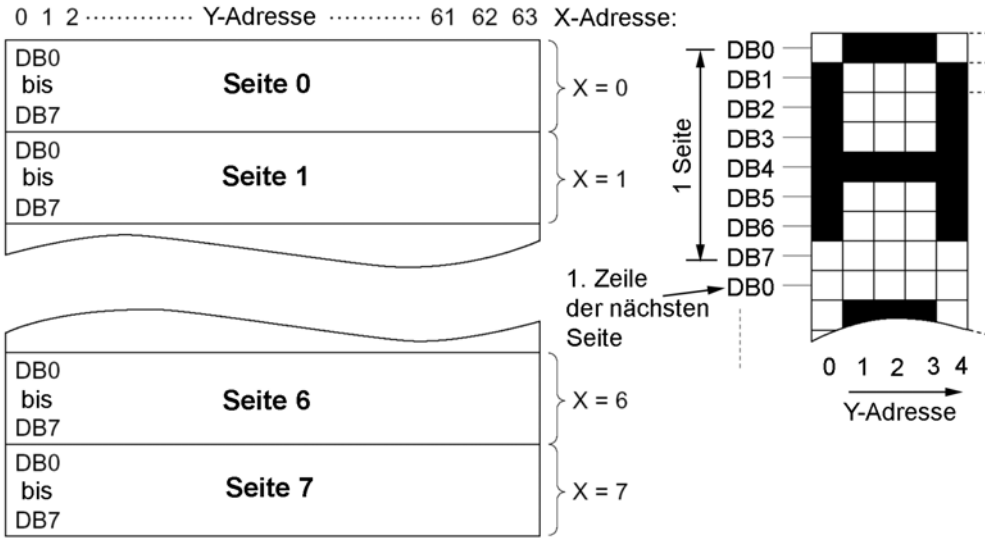

Abb. 29 Bildspeicherorganissation und Bildspeicheradressierung (nach Hitachi)

Start der Darstellung
Der Zeilenzähler bestimmt, mit welcher der maximal 64 Zeilen die Darstellung beginnt (Abb. 30). Er kann programmseitig geladen werden.

Anfangszustand nach dem Rücksetzen (Normalzustand, Abb. 30a):
Zeilenzähler = 0; Bildspeicherzeile 0 = 1. Zeile auf Bildschirm, Bildspeicherzeile 63 = 64. Zeile auf Bildschirm.

Darstellung mit Zeilenzähler ungleich Null (Abb. 30b...d). Erläuterung anhand von Abb. 30b:
Zeilenzähler = 1; Bildspeicherzeile 1 = 1. Zeile auf Bildschirm, Bildspeicherzeile 0 = 64. Zeile auf Bildschirm. Die oben abgeschnittenen Zeilen werden unten angehängt (Wrap Around).

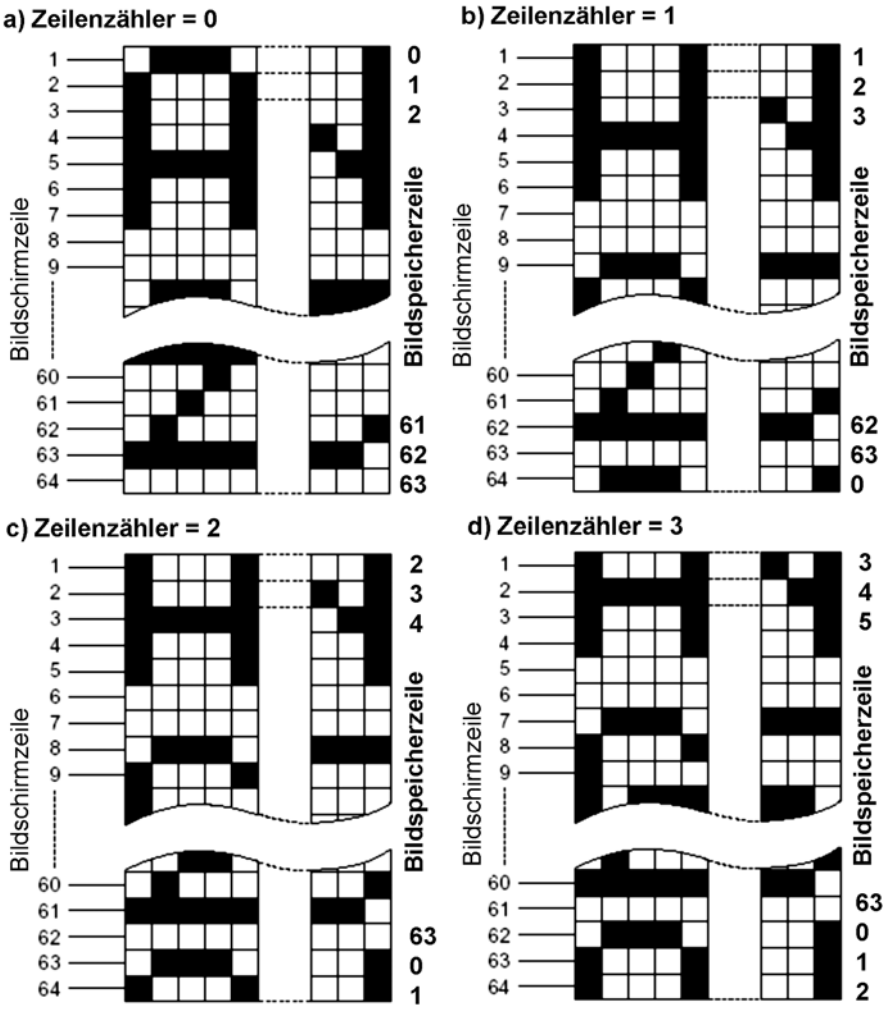

Abb. 30 Die Bilddarstellung in Abhängigkeit vom Zeilenzähler (nach Hitachi)

2.4 Kommandos

Kommando	Steuerl.		Datenbyte								Beschreibung
	D/I	R/W	7	6	5	4	3	2	1	0	
Display On/Off	0	0	0	0	1	1	1	1	1	D	Anzeige ein- oder ausschalten
Display Start Line	0	0	1	1	Zeilenzählerinhalt (0...63)						Zeilenzähler laden
Set Page (X Address)	0	0	1	0	1	1	1	X-Adresse (0...7)			X-Adresse (Seitenadresse) einstellen
Set Y Address	0	0	0	1	Y-Adresse (0...63)						Y-Adresse einstellen
Status Read	0	1	BF	0	D	R1	0	0	0	0	Zustand der Anzeige abfragen
Write Display Data	1	0	zu schreibendes Byte								Schreiben in Bildspeicher
Read Display Data	1	1	gelesenes Byte								Bildspeicher lesen

Tabelle 12 Kommandoübersicht

Display ON/OFF

D/I	R/W	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	1	D

Ein- und Ausschalten der Anzeige:
- D = 0: Anzeige ausgeschaltet. Daten bleiben erhalten.
- D = 1: Anzeige eingeschaltet.

Display Start Line

D/I	R/W	7	6	5	4	3	2	1	0
0	0	1	1	Zeilenzählerinhalt (0...63)					

Der Zeilenzähler wird mit der Belegung der Datenbits 5...0 geladen. Die betreffende Bildspeicherzeile wird zur ersten (obersten) Zeile auf dem Bildschirm (vgl. Abb. 30).

Set Page (X Address)

D/I	R/W	7	6	5	4	3	2	1	0
0	0	1	0	1	1	1	X-Adresse (0...7)		

Die X-Adresse (Seitenadresse) wird gemäß der Belegung der Datenbits 2...0 eingestellt.

Set Y Address

D/I	R/W	7	6	5	4	3	2	1	0
0	0	0	1	Y-Adresse (0...63)					

Die Y-Adresse (Spaltenadresse) wird gemäß der Belegung der Datenbits 5...0 eingestellt.

Status Read

D/I	R/W	7	6	5	4	3	2	1	0
0	1	BF	0	D	R	0	0	0	0

Lesen des Zustandes der Anzeige. Es sind nur die Bits 7, 5 und 4 von Bedeutung.
- BF = 0: Anzeige ist bereit, ein neues Kommando entgegenzunehmen.
- BF = 1: Anzeige ist mit der Ausführung eines Kommandos beschäftigt (Busy Flag). Kann nur Kommando Status Read ausführen. Kein anderes Kommando übertragen!
- D = 0: Anzeige ausgeschaltet.
- D = 1 : Anzeige eingeschaltet.
- R = 0: Anzeige betriebsfähig.
- R = 1: Anzeige ist mit Initialisierung beschäftigt (Rücksetzablauf). Kann nur Kommando Status Read ausführen. Kein anderes Kommando übertragen!

Write Display Data

D/I	R/W	7	6	5	4	3	2	1	0
1	0	Schreibdaten							

Schreiben eines Bytes in den Bildspeicher. Bildspeicheradresse ergibt sich aus Seiten- und Spaltenadresse (X , Y). Y-Adresse wird nach dem Schreiben automatisch um 1 erhöht.

Read Display Data

D/I	R/W	7	6	5	4	3	2	1	0
1	1	Lesedaten							

Lesen eines Bytes aus dem Bildspeicher. Bildspeicheradresse ergibt sich aus Seiten- und Spaltenadresse (X , Y). Y-Adresse wird nach dem Lesen automatisch um 1 erhöht.

Hinweis:
Nach dem Einstellen der Adresse ist zunächst ein Lesen ins Leere (Dummy Read) erforderlich. Erst danach kann das adressierte Byte gelesen werden.

2.5 Zugriffsabläufe

Die Bilddarstellung wird durch Folgen entsprechender Kommandos aufgebaut. Nach dem Senden eines Kommandos ist die Anzeige zunächst mit dessen Ausführung beschäftigt und kann kein weiteres Kommando annehmen (Besetztzustand). Kommandofolgen werden ebenso programmiert wie bei den Dotmatrixanzeigen (vgl. Abschnitt 1.5).

Wie lange dauert der Besetztzustand?
Es gibt keine eindeutigen Datenblattangaben. Die Dauer hängt von der jeweiligen Taktfrequenz ab, mit der der Schaltkreis betrieben wird (Takteingänge Φ1, Φ2; die Takterzeugung ist Sache der Anzeige):

- Dauer des Besetztzustandes: 1...3 Taktperioden.
- Dauer einer Taktperiode: 2,5...20 µs (Datenblattwerte).
- der Besetztzustand kann somit maximal 3 • 20 = 60 µs dauern.

Empfehlung:
Das Busy-Bit (BF) abfragen.

Initialisierung:

1. E auf 0. Datenbus auf Eingang (hochohmig).
2. Impuls (1 - 0 - 1) auf RST# geben (Dauer > 1 µs).
3. mit Kommandos Status Read das R-Bit abfragen. Wenn R = 0, kann normale Arbeit beginnen.
4. Bildspeicher mit Anfangsbelegung laden (z. B. löschen).
5. Bilddarstellung einschalten.

Bildspeicher füllen:

1. X-Adresse (Seitenadresse) einstellen.
2. ggf. anfängliche Y-Adresse einstellen.
3. Bytes schreiben. Y-Adresse zählt automatisch weiter (modulo 64).
4. Schritte 1 bis 3 solange wiederholen, bis Bilddarstellung übertragen ist.

Bildspeicherinhalt lesen:

1. X-Adresse (Seitenadresse) einstellen.
2. ggf. anfängliche Y-Adresse einstellen.
3. einmal Lesen ins Leere (Dummy Read).
4. Bytes lesen. Y-Adresse zählt automatisch weiter (modulo 64).
5. Schritte 1 bis 3 solange wiederholen, bis der gewünschte Speicherinhalt übertragen ist.

2.6 Vermischte Anregungen

Tip zur Programmorganisation
Im steuernden Controller eine Art Bildspeicher (als Array) vorsehen und wie einen Video-Bildspeicher betreiben. Anzeige bedarfsweise durch Blocktransporte aktualisieren.

Alternativen der Bildspeicherorganisation
Wir beziehen uns im folgenden auf eine Anzeige mit 128 • 64 Pixeln (zwei Controllerschaltkreise):

- als eindimensionales Array von 1024 Bytes. 1 Pixel = 1 Bit. Zugriffsfunktion der Art PIXEL (x, y, z); x, y = Bildkoordinaten, z = Pixelwert (0 oder 1). Funktion rechnet Koordinaten in Bitadressen um und ändert einzelne Bitwerte.
- als zweidimensionales Array (128 • 64) von 8 kBytes. 1 Pixel = 1 Byte. Arrayzugriffsvorkehrungen der Programmiersprache ausnutzen. Transportprogramm liest Array aus und errechnet daraus die zu übertragenden Bytes.

Wenn genug Platz ist:
Zwei Bildspeicher vorsehen (Prinzip Alt/Neu). Vor dem Transport jede einzelne Position überprüfen (Inhaltsvergleich), ob sich die Belegung geändert hat. Nur die geänderten Positionen müssen übertragen werden (eine Übertragung kostet um die 20...60 µs – da kann man schon etliche Bytes durchmustern...)

Bilddarstellung invertieren:
Den betreffenden Bildspeicherinhalt invertieren (Negation; XOR mit FFH).

Bilddarstellung blinken lassen:
Den betreffenden Bildspeicherinhalt einige Male je Sekunde zyklisch zwischen dem eigentlichen Inhalt und Bytes 00H oder dem invertierten Inhalt umladen. Blinkfrequenz 3... 10 Hz (ausprobieren).

Richtwerte zum programmseitigen Aktualisieren der Anzeige:

- 1 Byte erfordert 2,5 ...60 µs. Rechnen wir mit einem Mittelwert von 40 µs.
- 512 Bytes erfordern ≙ 20 ms.