Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
Lehrstuhl Rechnerarchitektur und -kommunikation

Praktikum **Computertechnik**

Versuch-Nr. 3: **Programmierbare Logik**

Einführung

Ziel des Versuches ist das Kennenlernen von Programmierbaren Logik-ICs. Verwendet werden hier das Multifunktionsgatter 4048 und der PAL16V8 (Programmed Array Logic).

Für die Programmierung des PAL16V8 wird die Software von Viewlogic und das Universalprogrammiergerät ALL03 verwendet.

Bei dieser Gelegenheit erhalten Sie einen Eindruck von CAE-Software (Computer Aided Engineering), die für die Simulation des PAL benutzt wird, aber ursprünglich für die Fertigung von ASICs (Application-Specific Integrated Circuit) und Prozessoren entwickelt wurde. Da das Erlernen der Bedienung für einen einzelnen Praktikumsversuch zu aufwendig ist, sind viele Teile vorbereitet.

Sie müssen sich nicht an die vorbereiteten Quellen halten, doch jede Abweichung wird von Ihnen einen höheren Zeitaufwand erfordern.

Für Interessenten kann vom Betreuer englischsprachige Literatur zur Einsicht bereitgestellt werden.

Versuchsvorbereitung

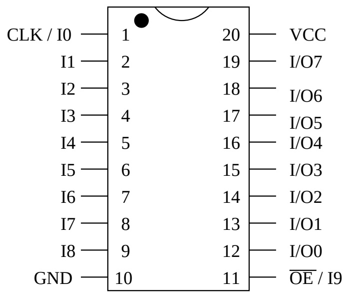
- Studium der Arbeitsblätter Nr. 1-4 (vor allem Nr. 3 Ablauf)
- Bekanntmachen mit dem Multifunktionsgatter 4048 (vgl. Anhang: Multifunktionsgatter) und dem PAL-Schaltkreis PAL16V8
- logisches Verhalten von Zähler-Schaltungen (Schaltbelegungstabelle)
- Aufstellung von Boolschen-Gleichungen

Schaltkreise für den Versuch 3 - Programmierbare Logik

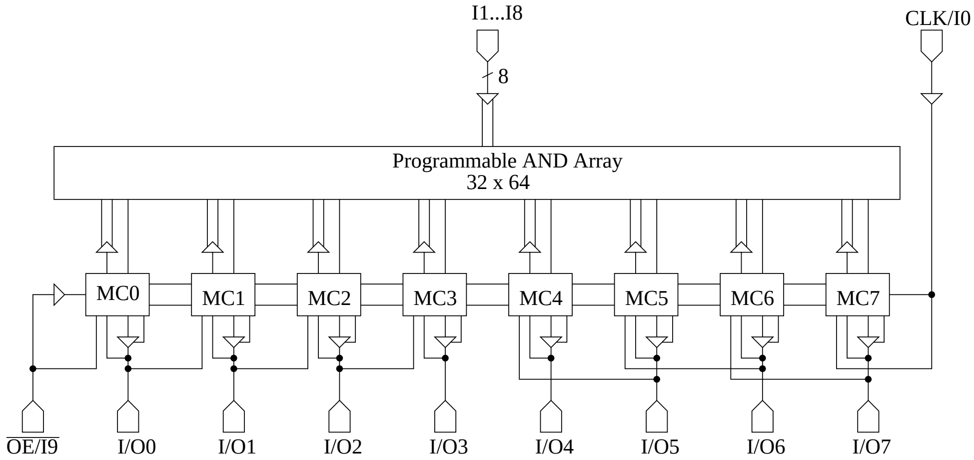
PALCE16V8

Der PALCE16V8 ist ein schneller, elektrisch löschbarer CMOS-Schaltkreis in einem 20 Pin DIP Gehäuse.

Der PALCE16V8 besteht aus einem programmierbaren AND-Array (Produktterme) und acht fixierten OR-Array (Summe). Jedes OR-Array läßt 8 Produktterme zu und speist eine Makrozelle, die jeweils ein Flip-Flop enthält. Eine anschauliche Darstellung der Schaltkreisstruktur befindet sich im Anhang.



Jede Ausgangs-Makrozelle kann unabhängig von den anderen Makrozellen sequentiell (Übergangsfunktion) oder kombinatorisch (Zuordner), high-aktiv oder low-aktiv programmiert werden.



Die Beschreibung des PAL für die Programmierung erfolgt durch Boolsche Gleichungen in der ABEL Hardware Description Language (ABEL-HDL). Der ABEL-Compiler erzeugt aus der Quelldatei in mehreren Schritten die Programmier- (JEDEC-) Datei.

JEDEC: Joint Electronic Device Engineering Council

Sequentielle Programmierung:

- Der Zuweisungsoperator in den Gleichungen ist :=
- Alle Flip-Flops in den 8 Makrozellen haben einen einzigen gemeinsamen Takteingang.
- Das bedeutet, daß die kombinatorischen Verknüpfungen der angelegten Daten nur während der steigenden Flanke des Taktes übernommen werden und bis zur nächsten steigenden Flanke des Taktes gültig bleiben, unabhängig davon wie sich die Eingabesignale verändern.
- Der Takteingang ist das PIN 1. Die Taktquelle muß in den Funktions-Gleichungen zugewiesen werden (z.B.: q.clk = clk).
- Die Tristate-Steuerung in allen Makrozellen erfolgt gemeinsam durch das PIN 11 (vgl. kombinatorische Programmierung).
- Das PIN 11 Output enable (/OE) ist low-aktiv und muß extern beschaltet sein (z.B. GND).

Tristate = dritte Zustand = wirkungsfrei = hochohmig

Kombinatorische Programmierung:

- Der Zuweisungsoperator in den Boolschen-Gleichungen ist =
- Alle Flip-Flops in den 8 Makrozellen haben einen einzigen gemeinsamen Takteingang.
- Das bedeutet, daß die kombinatorischen Verknüpfungen der angelegten Daten nur während der steigenden Flanke des Taktes übernommen werden und bis zur nächsten steigenden Flanke des Taktes gültig bleiben, unabhängig davon wie sich die Eingabesignale verändern.
- Bei der kombinatorischen Programmierung hängen die Ausgabedaten direkt von den Eingabedaten ab - ein Takt spielt keine Rolle.
- Es stehen nur 7 Produktterme zur Verfügung.
- Der 8. Produktterm jeder Makrozelle bestimmt deren Tristate-Steuerung
- Die Tristate-Steuerung kann in einem (dem 8.) Produktterm zugewiesen werden.
  z.B.: Y.OE = x0 & !x1 & x3 oder
  Y.OE = VCC (nie tristate)
- Das PIN 1 und 11 stehen als Eingang zur Verfügung (aber vgl. sequentielle Programmierung).
- Alle nicht benutzten Eingänge sollten beschaltet werden (VCC oder GND).

Arbeitsblatt Nr. 1

```
MODULE ABEL-Beispiel
TITLE    ' ABEL-Quelle als Beispiel
Autoren  : xyz, abc
Matrikel : ID 92
Datum    : 30.2.1792
'
DECLARATIONS
beispiel DEVICE 'p16v8';

" Festlegung der Schaltkreisanschlussbelegung
          I0,  I1,  I2,  I3,  I4,  I5,  I6,  I7,  I8, GND
PIN        1,   2,   3,   4,   5,   6,   7,   8,   9,  10;
        OEI9, IO0, IO1, IO2, IO3, IO4, IO5, IO6, IO7, VCC
PIN      11 ,  12,  13,  14,  15,  16,  17,  18,  19,  20;

" Zuordnung von Signalnamen
clk   = I0;     " Taktsignal
x0    = I7;     " kombinatorische Eingaenge
x1    = I8;
y1    = IO1;    " kombinatorische Ausgaenge
y2    = IO0;
q0    = IO7;    " sequentielle Ausgaenge
q1    = IO6;

EQUATIONS
" Beispiel fuer kombinatorische Programmierung: y1=(0,3)
y1 = !x0 & !x1
   #  x0 &  x1;

" weiteres kombinatorisches Beispiel: y2=(1,2):
y2 =  x0 & !x1
   # !x0 &  x1;

" 2-Bit Zaehler als Beispiel fuer sequentielle Programmierung
!q0 := q0;
!q1 :=  q0 &  q1
     # !q0 & !q1;

q0.clk  = clk;
q1.clk  = clk;

END ABEL-Beispiel
```

Arbeitsblatt Nr. 2

Auszug aus dem Syntax der ABEL Hardware Description Language zur Notation der Boolschen-Gleichungen

Bezeichner

Bezeichner müssen mit einem Buchstaben oder Unterstrich beginnen. Es können bis zu 30 große und kleine Buchstaben, Zahlen oder der Unterstrich folgen. Große und kleine Buchstaben werden unterschieden.

Leerzeichen sind die allgemeinen Trennzeichen.

Reservierte Schlüsselworte

CASE	EQUATIONS	MACRO	TEST_VECTORS
DECLARATIONS	FLAG	MODULE	THEN
DEVICE	FUSES	NODE	TITLE
ELSE	GOTO	OPTIONS	TRACE
ENABLE	IF	PIN	TRUTH_TABLE
END	IN	PROPERTY	WHEN
ENDCASE	ISTYPE	STATE	WITH
ENDWITH	LIBRARY	STATE_DIAGRAM	

Ausdrücke und Gleichungen

Ausdrücke und Gleichungen werden durch ein Semikolon abgeschlossen.

Operatoren

Operatoren	Beschreibung
!	logisches NOT
&	logisches AND
#	logisches OR
=	kombinatorische Zuweisung
:=	sequentielle (getaktete) Zuweisung
"	Beginn des Kommentars bis zum Zeilenende

Die logischen Operatoren in fallender Rangfolge sind: !, &, #

Format der Quelldatei

1. Vereinbarungsteil

MODULE *modul_id*	Bezeichner für den Namen des Designs
TITLE '*zeichenkette*'	Kurzbeschreibung zum Erkennen des Designs
DECLARATIONS	Schlüsselwort
device_id DEVICE *real_device*;	Festlegung des Schaltkreises
[!]*pin_id*[, [!]*pin_id* ...] PIN [*pin*#[, *pin*#]];	Beschreibung der Anschlüsse
id[, *id*] ... = *expr*[, *expr*] ... ;	Vereinbarung von Konstanten und Sets; Sets: Zusmf. von Signalen und Konstanten

Definitionen:

id	Bezeichner
expr	Ausdruck: Kombination von Bezeichnern und Operatoren
device_id	**DOS-Dateiname** ohne Erweiterung z.B. a2 (Aufgabe 2), der die Verbindung zwischen den Vereinbarungen und der Programmierdatei herstellt.
real_device	Zielschaltkreis; z.B. P16V8
pin_id	Bezeichner identifiziert jeden Anschluß am Schaltkreis-Symbol
pin#	Nummer des Schaltkreisanschlusses Eine *pin_id* und eine PIN-Nummer gehören immer zusammen.

Polarität

Wenn der Anschluß aktiv-low ist, kann mit einem führenden „!“ diese Signal-Aktivität festgelegt werden. Im Gegensatz dazu kann in den Gleichungen durch ein führendes „!“ festgelegt werden, daß das interne Signal inaktiv ist, wenn der Ausdruck *expr* wahr ist.

2. Zuweisungsteil

EQUATIONS	Schlüsselwort für diesen Abschnitt
[!]*element zuweisungsoperator expr*;	Boolsche Gleichung
signal_name.ext = *expr*;	Punkterweiterung
END *modul_id*;	Beendet die Quelle für das Design

Definitionen

element	Bezeichner für ein Signal oder ein Set von Signalen, dem der Wert des Ausdrucks zugewiesen wird
zuweisungsoperator	= für kombinatorische Zuweisungen := für sequentielle Zuweisungen
[!]	wenn *expr* wahr ist, wird *element* inaktiv; vgl. *pin_id*

Beispiel:

```
 y1     = !x0 & !x1  " kombinatorische Zuweisung
        #  x0 &  x1;
!q1     := q0 &  q1  " sequentielle Zuweisung
        # !q0 & !q1;
```

Punkterweiterungen von Signalen legen das Verhalten des Schaltkreises genauer fest.

.ext	Funktion
.CLK	Definiert die Taktquelle für alle sequentiellen Ausgänge. In diesem Praktikum ist der Bezeichner für den Takteingang clk einzusetzen.
.OE	Definiert die Freigabebedingung für die Tristate-Steuerung des Ausgangssignals in kombinatorischen Gleichungen (nicht nötig in diesem Praktikum).

Bei sequentiellen Gleichungen erfolgt die Tristate-Steuerung extern über den Freigabe-Anschluß.

Arbeitsblatt Nr. 3

Allgemeine Informationen zu den Softwarewerkzeugen

Die Werkzeuge arbeiten sowohl in einer UNIX als auch PC Umgebung. Das Programmiergerät muß von ipc227 im Praktikumsraum bedient werden.

Definition:

? im Dateinamen nimmt den Wert 2,3,41,42,43,44,45,46 oder 5 an. Dieser Wert ergibt sich aus der Aufgabenstellung, die Sie gerade durchführen. Z.B. wenn Sie die Aufgabe 42 durchführen, bezeichnet a?.* die Datei a42.* (Aufgabe 4.2).

Die Softwarewerkzeuge bestehen aus mehreren Teilen:

- TEXTEDIT: Ein ASCII-Texteditor
- VIEWDRAW: Eingabe und Bearbeitung eines Stromlaufplanes
- ABEL: ABEL-HDL Compiler zur Erzeugung eines Standard-JEDEC-Files aus einer ABEL-HDL Quelle
- VIEWSIM: Simulation
- VIEWTRACE: Darstellung von Signalverläufen
- ALL03: Programmierung des Schaltkreises

Damit Sie sich auf das Wesentliche konzentrieren können, wird die Durchführung der Aufgaben durch Skripte gesteuert, sind die Anschlußbelegung des Schaltkreises für die einzelnen Aufgaben (?.1), Quelldateien (a?.abl) und Stimuli (a?.cmd) vorbereitet. Die Quelldateien können mit jedem ASCII-Texteditor erzeugt werden.

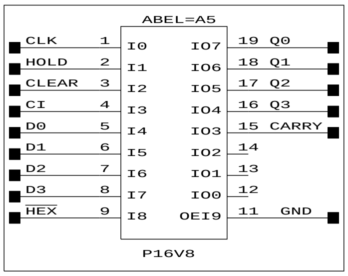
Ablauf

1. Eintragen der Boolschen Gleichungen in die Quelldatei

Das Skript wird in einem Fenster gestartet durch Eingabe von:

aufg?

Das sich öffnende VIEWDRAW-Fenster muß auf dem Bildschirm plaziert und eventuell in der Größe angepaßt werden. Es stellt die Schaltung für die Aufgabe ? dar. Anschließend öffnet sich ein Texteditorfenster, das ebenfalls auf dem Bildschirm zu plazieren und in der Größe anzupassen ist. Sie können darin die vorbereitete **Quelldatei editieren**.



Wenn Sie Ihre Boolschen Gleichungen in die Quelle eingetragen haben, **speichern** und **schließen** Sie bitte das Textfenster. Das Skript wird erst dann fortgesetzt.

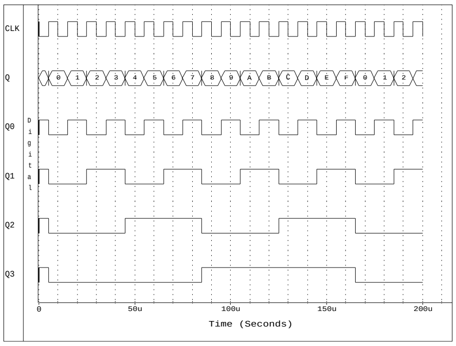
2. ABEL-HDL Compiler

Der Compiler führt eine Syntaxprüfung durch, optimiert die Funktionen und erzeugt eine JEDEC-**Programmierdatei** (p:\c2\v3\a?.jed). Mit dieser Datei kann der Schaltkreis später in dem Programmiergerät ALL03 programmiert werden. Weiterhin wird aus der JEDEC-Programmierdatei eine VHDL-Beschreibung der Schaltkreisfunktionen des PALCE16V8H erzeugt.

3. Die Simulation

Aus der VHDL-Schaltkreisbeschreibung erzeugt die Software von Viewlogic ein Simulationsmodel, das durch die Stimuli in der Datei aufg?.cmd angeregt und simuliert wird.

Das **Ergebnis** wird in dem VIEWTRACE-Fenster zusammen mit den Stimuli dargestellt und Sie können es durch Eingabe von **„w“** und Auswahl des Formates **„GENERIC“** in der Datei a?.gen **speichern**. Das Fenster sollte in seiner Größe so angepaßt werden, daß die Zahlen in den Signalverläufen von „q“ lesbar sind.



Falls das Ergebnis nicht der Aufgabenstellung entspricht,

- müssen die Gleichungen entsprechend geändert und gespeichert werden
- die Fenster TEXTEDIT, VIEWDRAW und VIEWTRACE geschlossen
- das Skript aufg? erneut aufgerufen werden

4. Programmierung des PAL

Die Programmierung des PAL erfolgt an dem Programmiergerät ALL03, der an den PC ipc227 angeschlossen ist. Die Programmerer-Software wird nach dem Anmelden des Benutzers automatisch gestartet. Der Ablauf für die Programmierung ist folgender:

1. Laden des JEDEC-Files **p:\c2\v3\a?.jed** mit dem Menüpunkt „**2.** Load file to buffer“ in das Programmiergerät (Es muß der **vollständige Pfad** angegeben werden.)
2. Einsetzen des zu programmierenden Schaltkreises in die Fassung des Programmiergerätes (Auf **richtige Lage der Anschlüsse** ist zu achten!)
3. Löschen des Schaltkreises (**E** Erase)
4. Blankcheck des Schaltkreises (**B** Blank check)
5. Programmierung des Schaltkreises (**P** Program)

5. Dokumentation der Simulation

Nach erfolgreicher Programmierung und Test des PAL können alle Fenster geschlossen und die nächste Aufgabe durchgeführt werden.

Am Ende der gesamten Versuchsdurchführung können für die Protokollierung die gespeicherten Simulationsergebnise a?.gen in ein VIETRACE-Fenster eingelesen werden, indem durch drücken der **Leertaste** eine Kommandoeingabe aufgerufen und durch Eingabe von **„mex sim?.vmc“** ein Makro gestartet wird. Durch Komandoeingabe von **„plotgo“** wird der Ausdruck ausgelöst.

Arbeitsblatt Nr. 4

Bedienungsanweisungen für die Logikanalyse

Sie benötigen das **I/O-, Display-** und **Print-Menü**, die durch drücken der entsprechenden Tasten angewählt werden können.

Jedes Menü besteht aus einer Reihe von Feldern, d.h. kleinen weiß oder grau eingerahmten Textboxen. Das gerade aktive Feld wird mit weißem Hintergrund dargestellt.

Zwischen den Menü-Feldern kann mit den Kursor-Tasten oder dem Drehknopf gewechselt werden.

Die Taste **SELECT** aktiviert bzw. deaktiviert ein Menü-Feld.

Grundeinstellung

1. **I/O -Menü** aufrufen
2. Menü-Feld **„LADEN“** aktivieren
3. In dem Menü **„Datei-Auswahl“** muß die gewünschte Datei (**A?.000**) mit den Kursor-Tasten angewählt und aktiviert werden.
4. Das Laden erfolgt, wenn das Bestätigungs-Feld **„V“** aktiviert wird. Das Abbruch-Feld **„X“** führt zum vorhergehenden Menü zurück.
5. Nun kann das **Display-Menü** gewählt werden. Die Grundeinstellung ist damit beendet.

Messung

Vor der Messung muß der Schaltkreis-Adapter auf den Schaltkreis aufgesetzt werden.

Damit ist der vorbereitete Anschluß des Logikanalysators erfolgt.

Die Messung wird gestartet durch betätigen der Taste **„RUN“**. Mit der Taste **„STOP“** kann die Messung abgebrochen werden. Die erfolgte Messung wird im Display-Menü sofort dargestellt, wenn die Triggerbedingung erfüllt ist.

Sie können sich nun die Messung genauer ansehen, indem sie das Feld hinter dem **X:** aktivieren (das ist das Feld für den X-Kursor) und mit dem Drehknopf durch die Messung fahren. Ganz rechts auf dem Display werden die Signal-Werte an der Position des X-Kursors angezeigt. Das Menü-Feld **„Z/Tlg:“** ermöglicht den Zeitmaßstab zu verändern.

Dokumentation

Die Dokumentation erfolgt mit dem Drucker. Sie brauchen lediglich die Taste **„PRINT“** zu drücken und es wird vom Drucker ein Bildschirmabzug gemacht.

Aufgabenstellungen

1. Multifunktionsgatter 4048

Es ist der Stromlaufplan mit nur einem einzigen Multifunktionsgatter 4048 ohne die Negation der Eingänge zur Realisierung einer der folgenden disjunktiven Normalformen anzugeben:

$$
\begin{aligned}
F(x3,\ldots,x0) = \quad Y &= \bigvee k_{jh}\{0,4,8\} \\
Y &= \bigvee k_{jh}\{7,\mathrm{d},\mathrm{f}\}
\end{aligned}
$$

Die Funktion der realisierten Schaltung ist nachzuweisen und zu diskutieren.

Durchführung:

1. Die minimierte Schaltung sollte in der Vorbereitung erarbeitet werden.
2. Die Schaltung ist mit dem Schaltkreis 4048 aufzubauen, die Funktion ist dem Betreuer nachzuweisen und als Schaltbelegungstabelle zu protokollieren.

2. Kombinatorische programmierbare Logik

Die in der 1. Aufgabe ausgeführte Verknüpfung ist mit dem PAL16V8 zu realisieren und zu diskutieren.

Durchführung:

1. Die Boolschen-Gleichungen sind vorzubereiten. (Siehe Arbeitsblatt Nr. 1 und 2)
2. Der PAL16V8 ist mit der Hilfe der vorhandenen Softwarewerkzeuge zu programmieren (siehe Arbeitsblatt Nr. 3)
3. Es ist eine Schaltung mit dem programmierten PAL16V8 aufzubauen.
4. Die Schaltung mit den dazugehörigen Gleichungen ist zu protokollieren und die Funktion ist dem Betreuer nachzuweisen.

3. Synchroner 4-Bit-Binärzähler

Für einen synchronen 4-Bit-Binärzähler sind die Boolschen-Gleichungen aufzustellen. Die Richtigkeit der Gleichungen ist nachzuweisen.

Durchführung:

1. Zur Vorbereitung sollte man sich Klarheit über die Verhaltensweise eines 4-Bit-Binärzählers verschaffen. Die Boolschen-Gleichungen sind aufzuschreiben.
2. Eine Schaltung zum Nachweis der Funktion ist mit einem programmierten PAL aufzubauen. (Die Schaltung kann zum Test langsam getaktet werden.) Zum Nachweis ist die Schaltung mit 100 kHz zu takten.
3. Zu protokollieren und zu diskutieren sind die Gleichungen, das Simultionsergebnis und die Logikanalyse (siehe Arbeitsblatt Nr. 4).

4. Synchroner 4-Bit-Binärzähler mit Steuerfunktion

Eine der folgenden Steuerfunktionen sind für einen synchronen 4-Bit-Binärzähler mit einem PAL16V8 zu realisieren:

1. Rückstellfunktion nach dem der Zählerstand 9 erreicht ist (Umschaltung zwischen hexadezimaler und dezimaler Zählsequenz)
2. Zähler mit einer Halte-Funktion
3. Zähler mit einer Rücksetz-Funktion
4. Zähler mit einer Lade-Funktion
5. Zähler mit einer Überlauf-Funktion
6. kaskadierbarer Zähler

Durchführung:

1. Klärung der Verhaltensweise eines Zählers mit der entsprechenden Steuerfunktion.
2. Eine Schaltung zum Nachweis der Funktion ist mit einem programmierten PAL aufzubauen. (Die Schaltung kann zum Test langsam getaktet werden.) Zum Nachweis ist die Schaltung mit 100 kHz zu takten.
3. Zu protokollieren und zu diskutieren sind die Gleichungen, das Simultionsergebnis und die Logikanalyse.

5. Zusatzaufgabe: Kaskadierbarer 4-Bit-Zähler mit mehreren Steuer-Funktionen

Realisierung eines kaskadierbaren synchronen 4-Bit-Binärzählers, der eine Rücksetz-, Halte- und Lade-Funktion mit Überlauf besitzt und zwischen hexadezimaler und dezimaler Zählsequenz umgeschaltet werden kann.

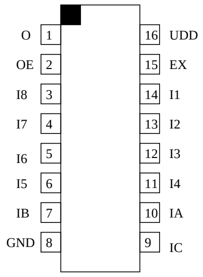
Der Funktionsnachweis ist wie in der Aufgabe 3 und 4 zu erbringen.

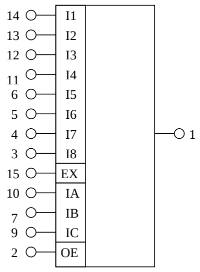
Anhang: Multifunktionsgatter 4048

Anschlußbelegung DIP 16, Schaltzeichen und IEC-Zeichen (in Entwicklung)

Bezeichnung der Anschlüsse:

I1...I8	Gattereingägnge
O	Ausgang
EX	Erweiterungseingang
GND	Bezugspotential
IA, IB, IC	Steuereingänge
UDD	Betriebsspannung
OE	Tristate-Zustandssteuerung des Ausganges





Der V 4048 D enthält ein programmierbares Multifunktionsgatter mit 8 Eingängen. Durck die 3 Steuereingägnge IA, IB, IC können 8 unterschiedliche Logikfunktionen festgelegt werden. Mit dem Steuereingang (OE) läßt sich der Tristate-Ausgang O steuern. Für OE = High liegt am Ausgang High oder Low, abhängig von den inneren Zuständen, für OE = Low befindet sich der Ausgang im hochohmigen Zustand. Über die 3 Steuereingägnge IA, IB, IC können folgende Funktionen (siehe Funktionstabelle) eingestellt werden: OR, NOR, AND, NAND, OR/AND, OR/NAND, AND/OR, AND/NOR. Über die 8 Logikeingänge I1...I8 hinaus hat der Schaltkreis einen Erweiterungseingang EX, mit dem eine Erweiterung der Anzahl der Logikeingänge möglich ist, z.B. durch einen weiteren V 4048 D auf 16 Eingänge. Wird der EX-Eingang nicht benutzt, so ist er auf GND zu legen.

Funktionstabelle (EX = L/OE = H)

Ausgangs-funktion	Logikfunktion	IA	IB	IC	unbenutzte Eingänge
NOR	O = NOT(I1+I2+I3+I4 + I5+I6+I7+I8)	L	L	L	GND
OR	O = I1+I2+I3+I4 + I5+I6+I7+I8	L	L	H	GND
OR/AND	O = (I1+I2+I3+I4)*(I5+I6+I7+I8)	L	H	L	GND
OR/NAND	O = NOT((I1+I2+I3+I4)*(I5+I6+I7+I8))	L	H	H	GND
AND	O = I1*I2*I3*I4 * I5*I6*I7*I8	H	L	L	VDD
NAND	O = NOT(I1*I2*I3*I4 * I5*I6*I7*I8)	H	L	H	VDD
AND/NOR	O = NOT((I1*I2*I3*I4)+(I5*I6*I7*I8))	H	H	L	VDD
AND/OR	O = (I1*I2*I3*I4)+(I5*I6*I7*I8)	H	H	H	VDD

Anhang: PALCE16V8 Logic Diagram