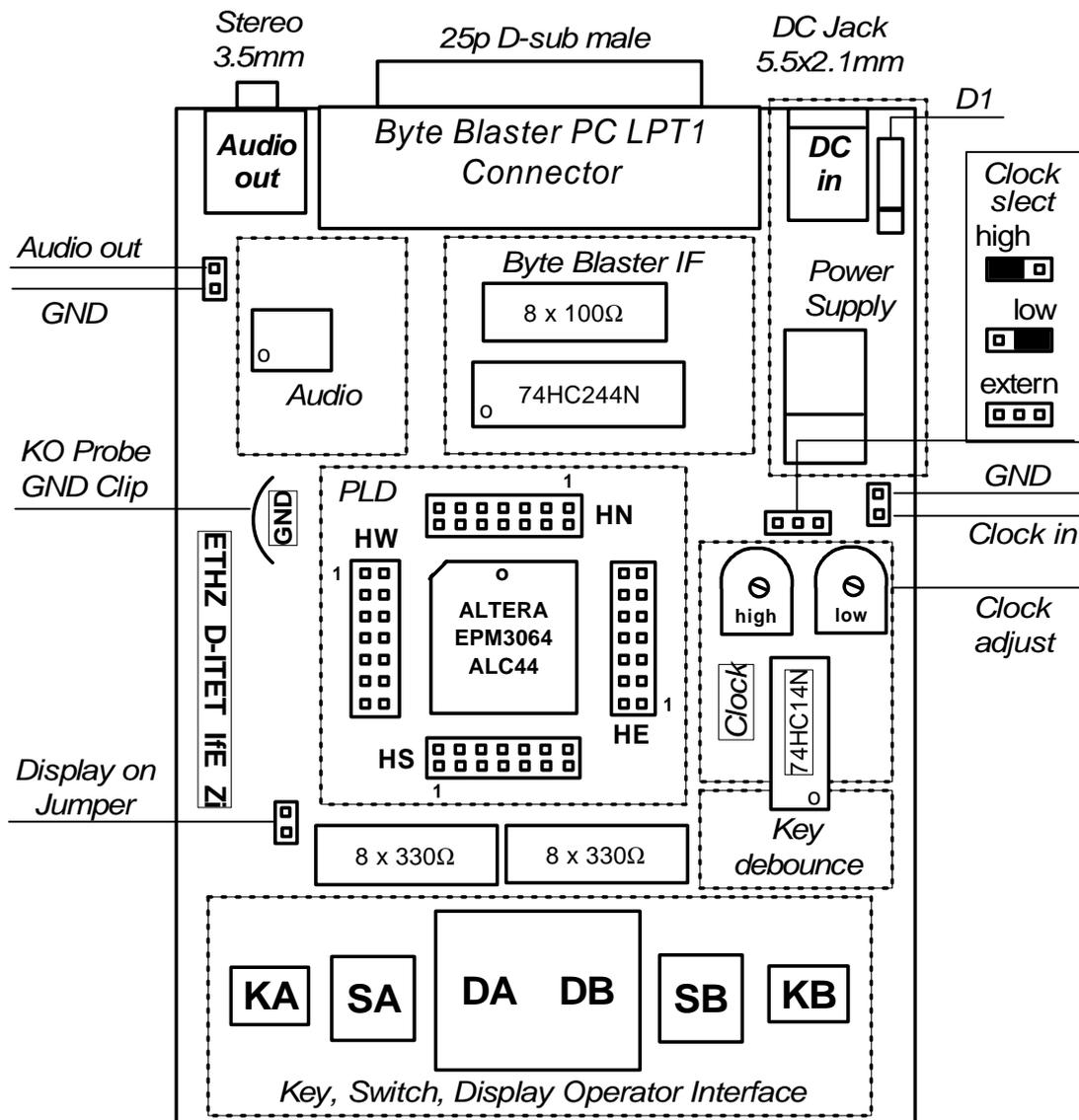


Informationen zum PLD Board SB2

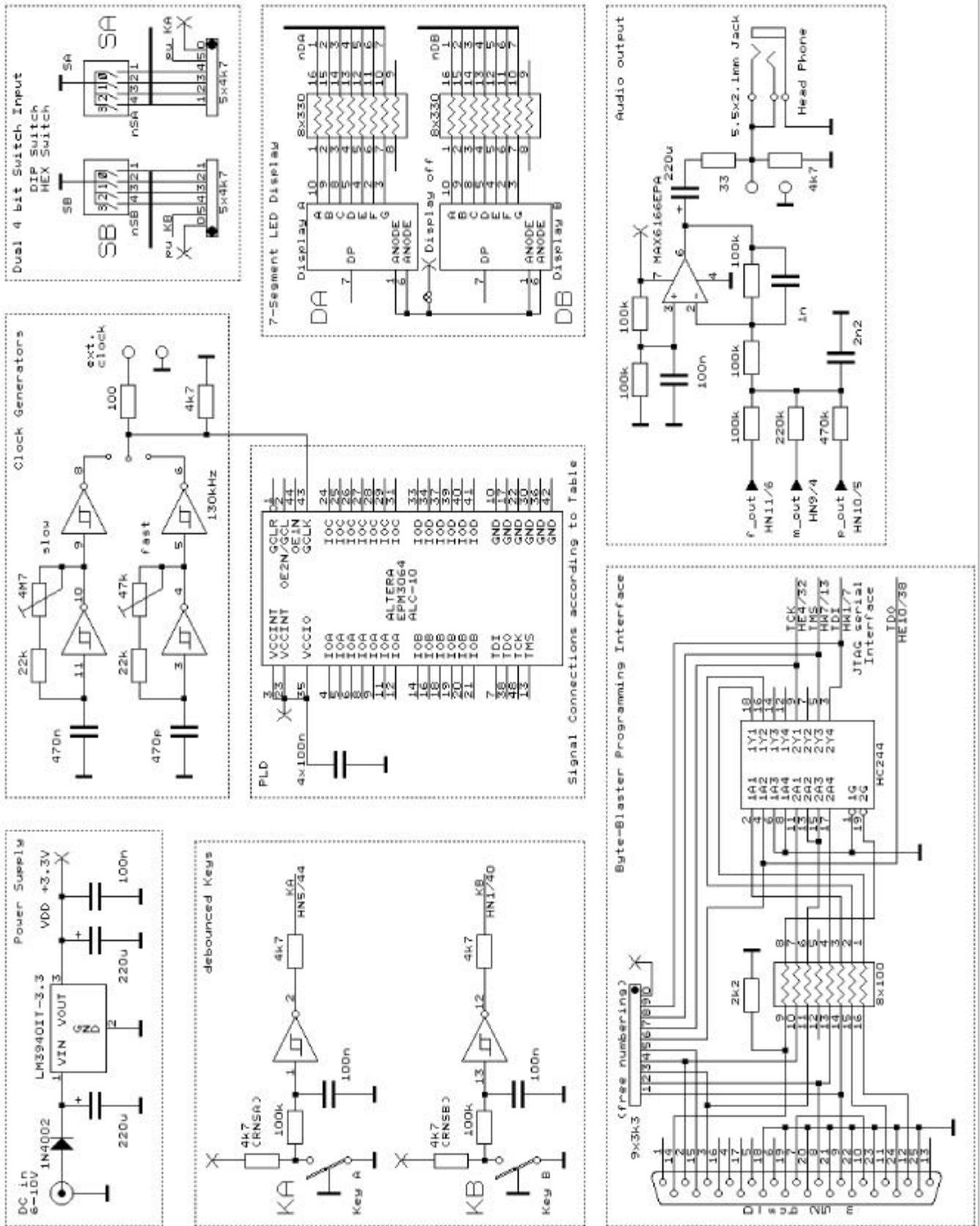
Bedienungselemente und Bauteile



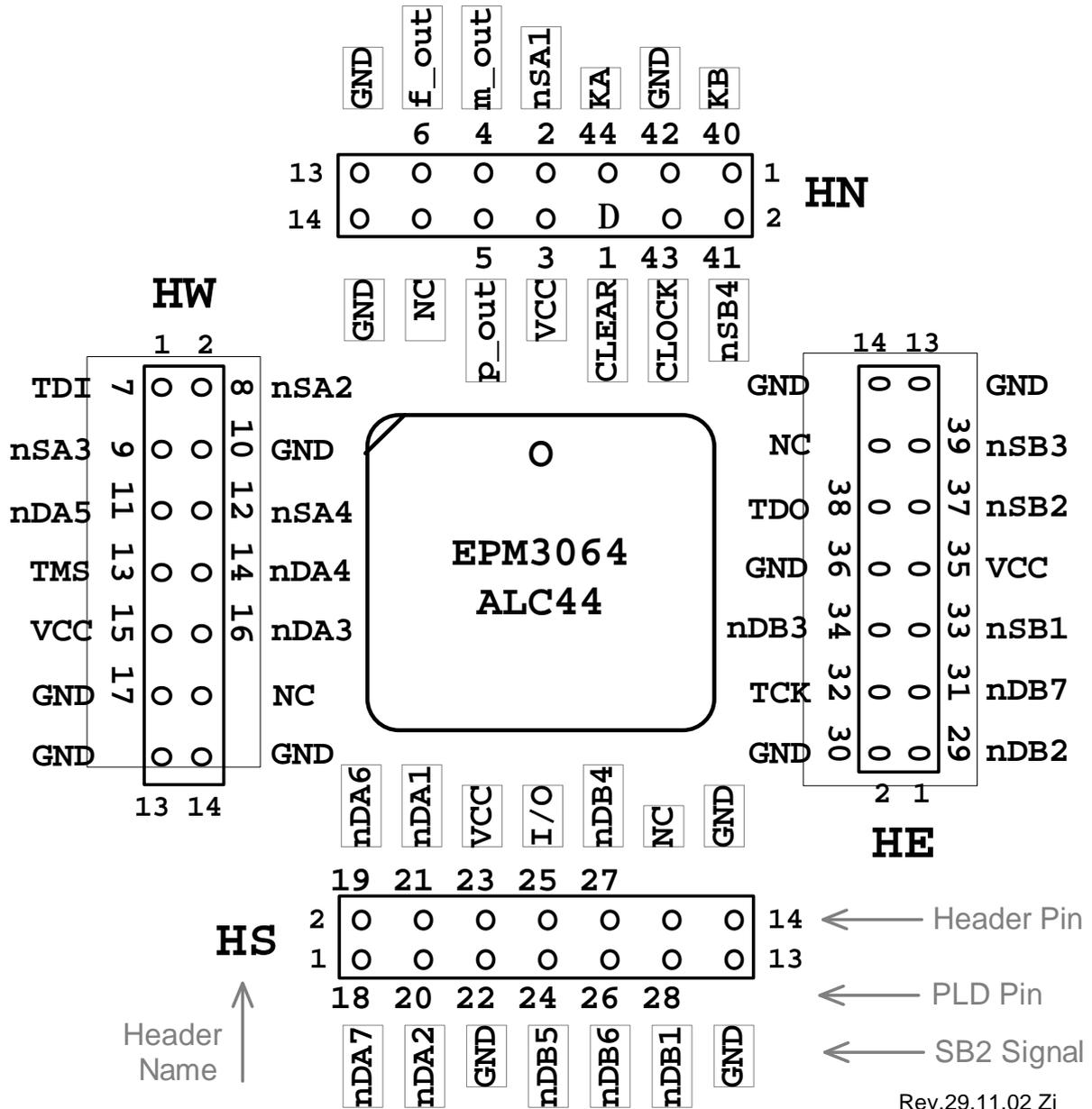
Byte Blaster Interface 8x100Ω zum Teil ersetzt durch 8x68Ω

Achtung: Wenn möglich zuerst die Speisung des Boards einschalten und nachher die Verbindung zum PC herstellen oder den PC einschalten!

Schaltschema



PLD Signale und Anschlüsse



Achtung: Für die globalen Signale Clock und Clear sollten in Design Files nur die folgenden von Altera vorgesehenen Bezeichnungen verwendet werden.

Clock => **GCLK1** (Global Clock 1)

Clear => **GCLRn** (Global Clear negative true)

Falls notwendig GCLK2 (Global Clock 2) über HN7 auf PLD-Pin 2 zuführen. Achtung: mit diesem Anschluss ist auch nSA1 verbunden, deshalb SA1 unbedingt auf off stellen.

In der Tabelle auf der nächsten Seite sind alle Signale und Anschlüsse des PLDs nach verschiedenen Kriterien sortiert zusammengestellt.

| Sortiert nach Signal | | | Sortiert nach Header | | | Sortiert nach IC Pin | | |
|----------------------|-----------------|------------|----------------------|-----------------|------------|----------------------|-----------------|------------|
| Signal SB2 | 14pin Header | PLCC 44 | Signal SB2 | 14pin Header | PLCC 44 | Signal SB2 | 14pin Header | PLCC 44 |
| Clear | HN6 | 1 | nDB2 | HE1 | 29 | Clear | HN6 | 1 |
| Clock | HN4 | 43 | GND | HE2 | 30 | nSA1 | HN7 | 2 |
| f_out | HN11 | 6 | nDB7 | HE3 | 31 | VCC | HN8 | 3 |
| GND | HE2 | 30 | TCK | HE4 | 32 | m_out | HN9 | 4 |
| GND | HE8 | 36 | nSB1 | HE5 | 33 | p_out | HN10 | 5 |
| GND | HN3 | 42 | nDB3 | HE6 | 34 | f_out | HN11 | 6 |
| GND | HS5 | 22 | VCC | HE7 | 35 | TDI | HW1 | 7 |
| GND | HW11 | 17 | GND | HE8 | 36 | nSA2 | HW2 | 8 |
| GND | HW4 | 10 | nSB2 | HE9 | 37 | nSA3 | HW3 | 9 |
| I/O | HS8 | 25 | TDO | HE10 | 38 | GND | HW4 | 10 |
| KA | HN5 | 44 | nSB3 | HE11 | 39 | nDA5 | HW5 | 11 |
| KB | HN1 | 40 | | HE12 | NC | nSA4 | HW6 | 12 |
| m_out | HN9 | 4 | | HE13 | GND | TMS | HW7 | 13 |
| nDA1 | HS4 | 21 | | HE14 | GND | nDA4 | HW8 | 14 |
| nDA2 | HS3 | 20 | KB | HN1 | 40 | VCC | HW9 | 15 |
| nDA3 | HW10 | 16 | nSB4 | HN2 | 41 | GND | HW11 | 17 |
| nDA4 | HW8 | 14 | GND | HN3 | 42 | nDA7 | HS1 | 18 |
| nDA5 | HW5 | 11 | Clock | HN4 | 43 | nDA6 | HS2 | 19 |
| nDA6 | HS2 | 19 | KA | HN5 | 44 | nDA2 | HS3 | 20 |
| nDA7 | HS1 | 18 | Clear | HN6 | 1 | nDA1 | HS4 | 21 |
| nDB1 | HS11 | 28 | nSA1 | HN7 | 2 | GND | HS5 | 22 |
| nDB2 | HE1 | 29 | VCC | HN8 | 3 | VCC | HS6 | 23 |
| nDB3 | HE6 | 34 | m_out | HN9 | 4 | nDB5 | HS7 | 24 |
| nDB4 | HS10 | 27 | p_out | HN10 | 5 | I/O | HS8 | 25 |
| nDB5 | HS7 | 24 | f_out | HN11 | 6 | nDB6 | HS9 | 26 |
| nDB6 | HS9 | 26 | | HN12 | NC | nDB4 | HS10 | 27 |
| nDB7 | HE3 | 31 | | HN13 | GND | nDB1 | HS11 | 28 |
| nSA1 | HN7 | 2 | | HN14 | GND | nDB2 | HE1 | 29 |
| nSA2 | HW2 | 8 | nDA7 | HS1 | 18 | GND | HE2 | 30 |
| nSA3 | HW3 | 9 | nDA6 | HS2 | 19 | nDB7 | HE3 | 31 |
| nSA4 | HW6 | 12 | nDA2 | HS3 | 20 | TCK | HE4 | 32 |
| nSB1 | HE5 | 33 | nDA1 | HS4 | 21 | nSB1 | HE5 | 33 |
| nSB2 | HE9 | 37 | GND | HS5 | 22 | nDB3 | HE6 | 34 |
| nSB3 | HE11 | 39 | VCC | HS6 | 23 | VCC | HE7 | 35 |
| nSB4 | HN2 | 41 | nDB5 | HS7 | 24 | GND | HE8 | 36 |
| p_out | HN10 | 5 | I/O | HS8 | 25 | nSB2 | HE9 | 37 |
| TCK | HE4 | 32 | nDB6 | HS9 | 26 | TDO | HE10 | 38 |
| TDI | HW1 | 7 | nDB4 | HS10 | 27 | nSB3 | HE11 | 39 |
| TDO | HE10 | 38 | nDB1 | HS11 | 28 | KB | HN1 | 40 |
| TMS | HW7 | 13 | | HS12 | NC | nSB4 | HN2 | 41 |
| VCC | HE7 | 35 | | HS13 | GND | GND | HN3 | 42 |
| VCC | HN8 | 3 | | HS14 | GND | Clock | HN4 | 43 |
| VCC | HS6 | 23 | TDI | HW1 | 7 | KA | HN5 | 44 |
| VCC | HW9 | 15 | nSA2 | HW2 | 8 | nDA3 | HW10 | 16 |
| | HN12 | NC | nSA3 | HW3 | 9 | | HN13 | GND |
| | HN13 | GND | GND | HW4 | 10 | | HN14 | GND |
| | HN14 | GND | nDA5 | HW5 | 11 | | HS13 | GND |
| | HS12 | NC | nSA4 | HW6 | 12 | | HS14 | GND |
| | HS13 | GND | TMS | HW7 | 13 | | HW13 | GND |
| | HS14 | GND | nDA4 | HW8 | 14 | | HW14 | GND |
| | HW12 | NC | VCC | HW9 | 15 | | HE13 | GND |
| | HW13 | GND | nDA3 | HW10 | 16 | | HE14 | GND |
| | HW14 | GND | GND | HW11 | 17 | | HN12 | NC |
| | HE12 | NC | | HW12 | NC | | HS12 | NC |
| | HE13 | GND | | HW13 | GND | | HW12 | NC |
| | HE14 | GND | | HW14 | GND | | HE12 | NC |

Signal zu Pin Zuordnungsfile für MAX+plusII

Die Zuordnung der logischen Signale zu den physikalischen Pins des PLD, das Pin-Locking, wird für jedes Projekt im .acf File definiert. Aus dem SB2_Pin-Lock-List File können entweder alle Definitionen (führt zu Warnungen für jedes effektiv im Projekt nicht verwendete Signal, dafür vergisst man sicher keines) oder nur die tatsächlich verwendeten ins acf File kopiert werden. PLD Typ (DEVICE - Zeile) und Chipname (CHIP – Zeile) müssen immer ins acf File übernommen werden. Falls der 2. globale Clock benützt wird, muss GCLK2 anstatt nSA1 auf PIN 2 gelockt werden.

SB2_Pin-Lock-List.txt:

```
-- Pin lock list for digiprakt SB2 board
-- 15.10.02 Zi
-- copy into project .adf file at the beginning
-- (just after the copyright comment sermon,
-- before the default devices section).
-- you can delete unused pins form the list
-- to avoid warnings during compilation.
-- commented out unused pins are deleted
-- during the next compilation!
-- prefix n indicates negative true signal
-- do not forget to edit the chip name
```

```
CHIP      MySB2ChipName
BEGIN
  DEVICE = EPM3064ALC44-10;
  GCLK1 : INPUT_PIN = 43;
  m_out : OUTPUT_PIN = 4;
  f_out  : OUTPUT_PIN = 6;
  p_out  : OUTPUT_PIN = 5;
  KA : INPUT_PIN = 44;
  KB : INPUT_PIN = 40;
  nSB1 : INPUT_PIN = 33;
  nSB2 : INPUT_PIN = 37;
  nSB3 : INPUT_PIN = 39;
  nSB4 : INPUT_PIN = 41;
  nDB1 : OUTPUT_PIN = 28;
  nDB2 : OUTPUT_PIN = 29;
  nDB3 : OUTPUT_PIN = 34;
  nDB4 : OUTPUT_PIN = 27;
  nDB5 : OUTPUT_PIN = 24;
  nDB6 : OUTPUT_PIN = 26;
  nDB7 : OUTPUT_PIN = 31;
  nDA1 : OUTPUT_PIN = 21;
  nDA2 : OUTPUT_PIN = 20;
  nDA3 : OUTPUT_PIN = 16;
  nDA4 : OUTPUT_PIN = 14;
  nDA5 : OUTPUT_PIN = 11;
  nDA6 : OUTPUT_PIN = 19;
  nDA7 : OUTPUT_PIN = 18;
  nSA1 : INPUT_PIN = 2;
  nSA2 : INPUT_PIN = 8;
  nSA3 : INPUT_PIN = 9;
  nSA4 : INPUT_PIN = 12;
END;
```

Wichtige Punkte der MAX+plusII Anwendung

Signalnamen:

Maximal 32 Alphanumerische Zeichen und Underline zulässig.

Signalnamen mit Grossbuchstaben beginnen.

Für negativ true (aktiv L, invertiert) Signale Präfix n verwenden (z.B nSB1).

Namen von einzelnen Signalen nicht mit einer Zahl abschliessen.

Signale in einer Gruppe mit Zahl am Schluss kennzeichnen, z.B. nDB1, nDB2 nDB7 und nicht nDBa, nDBb, nDBc.... Dadurch wird es möglich, diese gesamthaft als Bus einzusetzen, z.B. nDB[1..7].

Graphik Editor, *.gdf Files

Objekte Selektieren (Zum Bearbeiten, Kopieren, Löschen, Verschieben):

Einzelnes Objekt: Linksclick ins Objekt, roter Begrenzungsrahmen erscheint

Mehrere Objekte: Rahmen um die Objekte aufziehen (mit gedrückter Linkstaste ab freiem Punkt in der Zeichnungsebene einziehen, nur vollständig innerhalb des Rahmens liegende Objekte werden selektiert, Leitungen nur bis zum Rahmen.

Kopieren: Ctrl Drag&Drop, natürlich auch Ctrl C, Ctrl V oder Edit Menü.

Verschieben: Verbindungen nicht trennen, Options > Rubberbanding = on

Verbindungen trennen: Options > Rubberbanding = off

Altera Hardward Description Language, AHDL *.tdf Files

Logische Operatoren:

NOT (!), AND (&), NAND (!&), OR (#), NOR (!#), XOR (\$), and XNOR (!\$).

Subdesign Name zwingend gleich Filenamen, genau 1 Subdesign pro File.

Kommentare in % eingeschlossen, innerhalb einer Zeile und über mehrere Zeilen möglich. Beispiel % Kommentar %. Der MAX+ Text-Editor hebt Kommentare farblich hervor. Praktisch: Er lässt sich mit der Insert-Taste von Insert auf Overwrite Modus umschalten (.Anzeige in unterem Fensterrand)

Vorlagen für die Beschreibung mit logischen Gleichungen und Wahrheitstabellen verwenden, wir beschränken uns darauf.

Graphik und AHDL Design Files

Speichern und Fehlercheck:

File zum aktuellen Projekt machen, File > Project > SetProjectToCurrentFile,

File > Project > Save&Check

Fehler müssen behoben werden, Warnungen nicht.

Compilieren: MAX+plusII > Compiler

Programmieren: MAX+plusII > Programmer (pof File des aktuellen Projektes)

Hierarchie anzeigen:MAX+plusII > HierarchyDisplay

Archivieren: File > Project > Archive

Archiviert alles in der Hierarchie nach unten ab dem aktuellen Projekt File in einem Archivverzeichnis.

Umbenennen von Signalnamen und Kommentar:

Utilities > Search&Replace

Der PLD Altera EPM3064

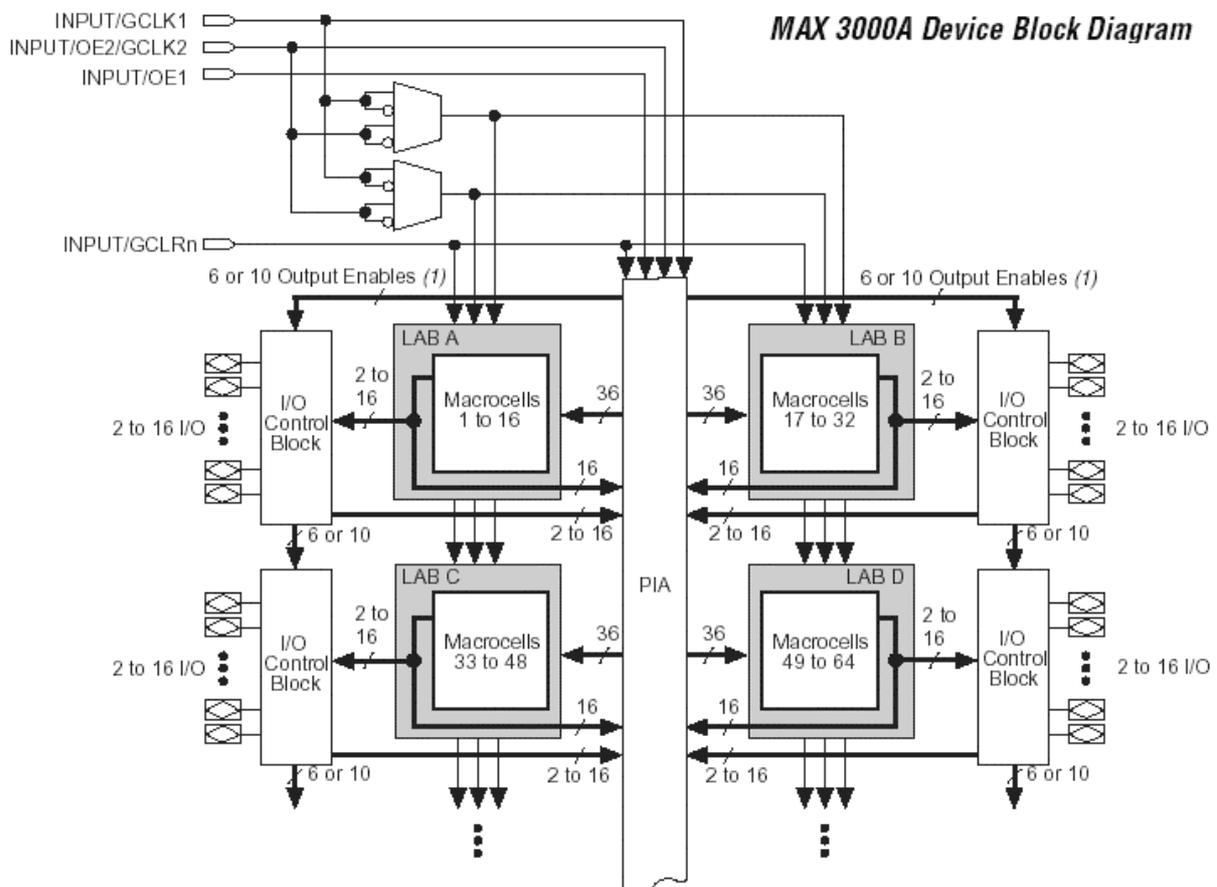
Die Informationen in diesem Abschnitt sind nur für besonders interessierte Studierenden gedacht, Sie müssen diese wirklich nicht studieren.

Unser PLD – IC ist ein Baustein aus der aktuellen Altera MAX3000A Familie. Er wird mit 3.3V Speisespannung betrieben, die Programmierung erfolgt nicht flüchtig (bleibt beim Ausschalten der Speisung bestehen) mit Floating-Gate Feldeffekt Transistoren (wie in EEPROM und Flash-Speichern). Er kann beliebig oft neu programmiert werden (ist elektrisch löscher), dazu ist keine Programmiergerät notwendig, alle dazu nötigen Schaltungen sind auf dem IC mitintegriert. Die EPM3000 Familie ist (wie fast alle moderneren PLDs) 'on board' über eine genormte serielle 'JTAG' Schnittstelle programmierbar. Der 3064 enthält 64 PAL-Zellen (Macrocells) die in 4 Blöcken (LABs, Logic Array Blocks) zu je 16 organisiert sind. Jeder dieser Blöcke ist in der Komplexität etwa vergleichbar mit einem GAL (Generic Array Logic IC). Die weiteren Mitglieder der 3000 Familie unterscheiden sich nur in der Anzahl der Blöcke (2 bis 16). Die folgenden Bilder zeigen den wesentlichen Aufbau.

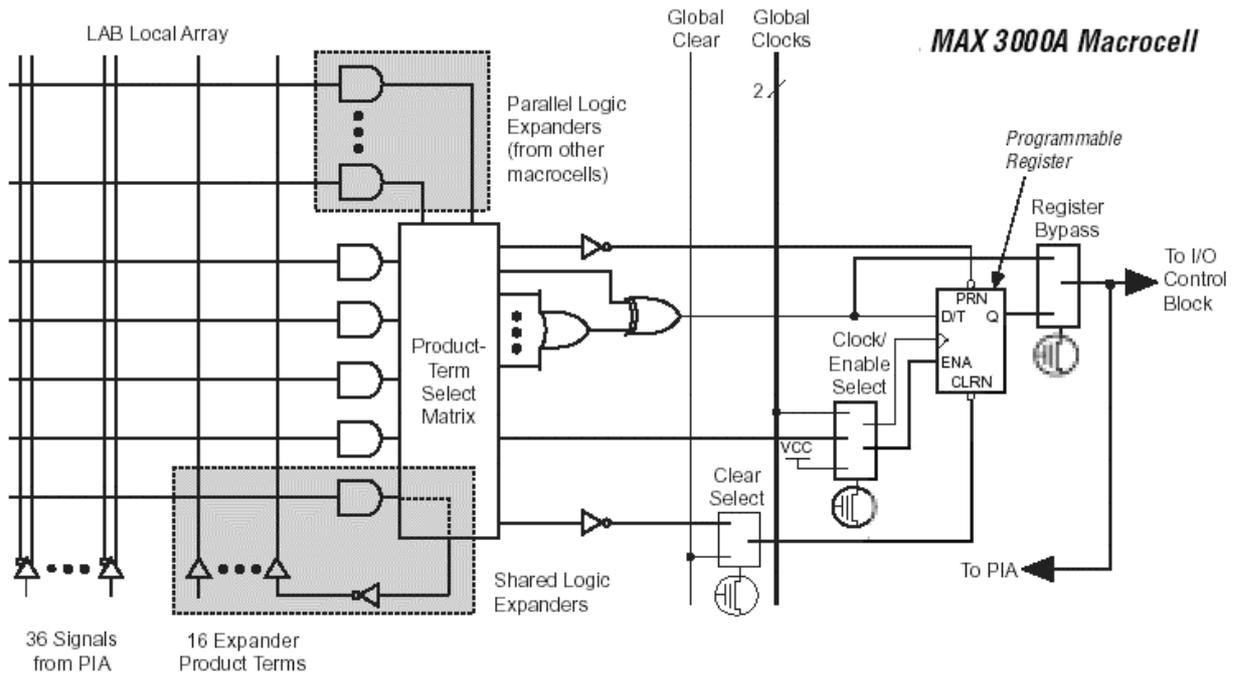
Das vollständige Datenblatt der MAX3000 Familie kann von der Praktikumsseite oder von Altera.com heruntergeladen werden (PDF, ca 700k).

Ein sehr gutes Tutorial (Folien einer Präsentation) über Altera PLDs und MAX+plusII steht ebenfalls auf der Praktikumsseite zur Verfügung (PDF, ca 3M).

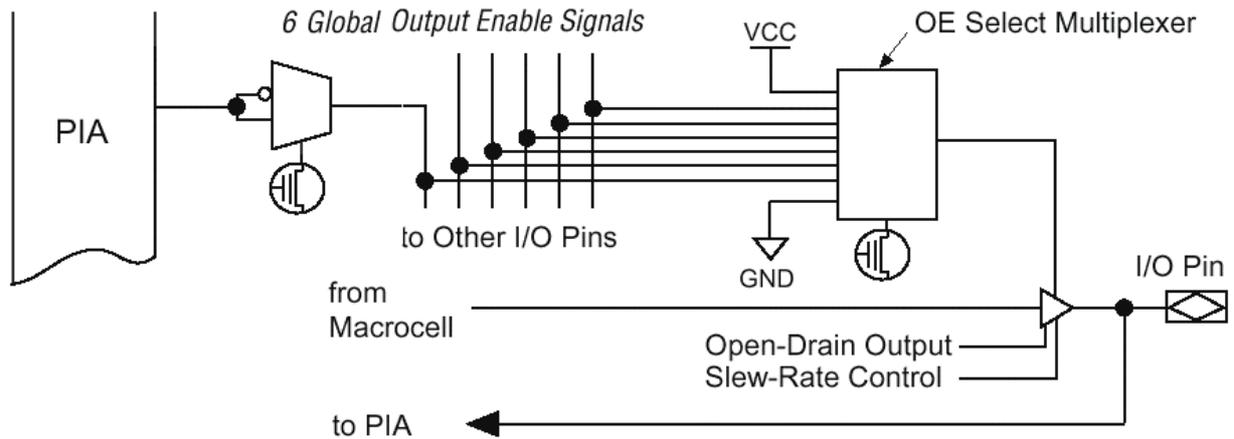
Die Altera Digital Library CD enthält umfassende Informationen und Software.



Architektur MAX3000 PLD Familie



Architektur MAX3000 Makrozelle (Logikzelle, PAL Zelle)



Altera EPM3064A I/O Pin Control Block

Altera EPM3064A Pin and SB2 Signal Table

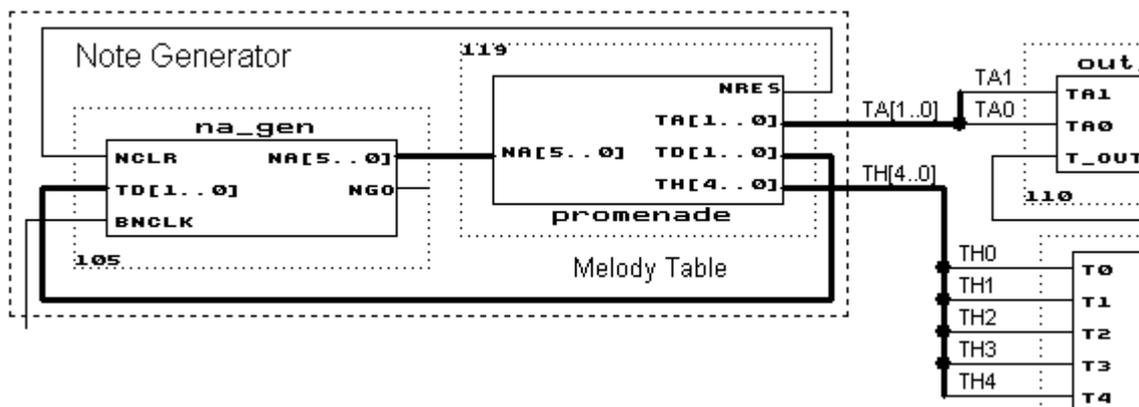
Die Tabelle auf der folgenden Seite zeigt für alle EPM3064 - Pins deren charakteristischen Eigenschaften: Art des Pins (z.B I/O, GND, VCC usw.), die Zuordnung zu den 4 Logikblöcken (Logic Array Blocks LAB A, B, C, D) und den Logikzellen (Macrocells, LCells) sowie die Verfügbarkeit globaler Signale in den einzelnen Logikzellen.

| PLD Function | PLCC 44 Pin | JTAG | LCell | OE MUX Pin/LCell | LAB | SB2 Signal | 14pin Header |
|------------------|-------------|------|-------|------------------|-----|------------|--------------|
| Input/GCLRn | 1 | - | - | -/- | - | NC | HN6 |
| Input/OE2n/GCLK2 | 2 | - | - | -/- | - | nSA1 | HN7 |
| VCCINT | 3 | - | - | -/- | - | VCC | HN8 |
| I/O or Buried | 4 | - | 16 | 4/- | A | m_out | HN9 |
| I/O or Buried | 5 | - | 14 | 5/- | A | p_out | HN10 |
| I/O or Buried | 6 | - | 11 | 3/1 | A | f_out | HN11 |
| I/O or Buried | 7 | TDI | 8 | -/- | A | TDI | HW1 |
| I/O or Buried | 8 | - | 5 | 6/4 | A | nSA2 | HW2 |
| I/O or Buried | 9 | - | 4 | -/3 | A | nSA3 | HW3 |
| GND | 10 | - | - | -/- | - | GND | HW4 |
| I/O or Buried | 11 | - | 3 | 1/2 | A | nDA5 | HW5 |
| I/O or Buried | 12 | - | 1 | -/2 | A | nSA4 | HW6 |
| I/O or Buried | 13 | TMS | 32 | -/- | B | TMS | HW7 |
| I/O or Buried | 14 | - | 30 | 3/1 | B | nDA4 | HW8 |
| VCCIO | 15 | - | - | -/- | - | VCC | HW9 |
| I/O or Buried | 16 | - | 25 | 5/6 | B | nDA3 | HW10 |
| GND | 17 | - | - | -/- | - | GND | HW11 |
| I/O or Buried | 18 | - | 21 | -/2 | B | nDA7 | HS1 |
| I/O or Buried | 19 | - | 20 | -/5 | B | nDA6 | HS2 |
| I/O or Buried | 20 | - | 19 | 6/4 | B | nDA2 | HS3 |
| I/O or Buried | 21 | - | 17 | 6/5 | B | nDA1 | HS4 |
| GND | 22 | - | - | -/- | - | GND | HS5 |
| VCCINT | 23 | - | - | -/- | - | VCC | HS6 |
| I/O or Buried | 24 | - | 33 | -/5 | C | nDB5 | HS7 |
| I/O or Buried | 25 | - | 35 | 6/4 | C | NC | HS8 |
| I/O or Buried | 26 | - | 36 | -/5 | C | nDB6 | HS9 |
| I/O or Buried | 27 | - | 37 | -/- | C | nDB4 | HS10 |
| I/O or Buried | 28 | - | 40 | 1/3 | C | nDB1 | HS11 |
| I/O or Buried | 29 | - | 41 | 5/6 | C | nDB2 | HE1 |
| GND | 30 | - | - | -/- | - | GND | HE2 |
| I/O or Buried | 31 | - | 46 | 2/1 | C | nDB7 | HE3 |
| I/O or Buried | 32 | TCK | 48 | -/- | C | TCK | HE4 |
| I/O or Buried | 33 | - | 49 | -/2 | D | nSB1 | HE5 |
| I/O or Buried | 34 | - | 51 | 1/2 | D | nDB3 | HE6 |
| VCCIO | 35 | - | - | -/- | - | VCC | HE7 |
| GND | 36 | - | - | -/- | - | GND | HE8 |
| I/O or Buried | 37 | - | 53 | 6/4 | D | Nsb2 | HE9 |
| I/O or Buried | 38 | TDO | 56 | -/- | D | TDO | HE10 |
| I/O or Buried | 39 | - | 57 | 3/- | D | nSB3 | HE11 |
| I/O or Buried | 40 | - | 62 | 5/6 | D | KB | HN1 |
| I/O or Buried | 41 | - | 64 | 4/- | D | nSB4 | HN2 |
| GND | 42 | - | - | -/- | - | GND | HN3 |
| Input/GCLK1 | 43 | - | - | -/- | - | Clock | HN4 |
| Input/OE1n | 44 | - | - | 1/- | - | KA | HN5 |

Altera EPM3064A Pin and SB2 Signal Table

Hinweis zur Bezeichnung von Bussignalen in MAX+plusII:

Für Busse (Signalvektoren) gelten besondere Vorschriften für die Bezeichnung: zwischen Symbolen in denen die Input-Output Bussignale identisch bezeichnet sind, können Busverbindungen direkt ohne weitere Bezeichnung gezogen werden (NA[5..0], TD[1..0] im folgenden Bild). Sobald die Bezeichnungen in den Symbolen nicht mehr identisch sind, oder ein Bus in einzelne Signale aufgeteilt wird, müssen sowohl der Bus als auch die einzelnen Signale beschriftet werden (TH[4..0] im folgenden Bild). Die explizite Beschriftung ist auch dann notwendig, wenn die Symbolanschlüsse im Prinzip kompatible Namen aufweisen (wie TA[1..0] und TA1, TA0 im folgenden Bild). Manual und Help stellen dies leider nur teilweise richtig dar. Signalnamen müssen immer über einer Leitung oder einem Bus positioniert werden.



Beispiel für die Bezeichnung von Bus-Signalen

Schlechter und guter Stil

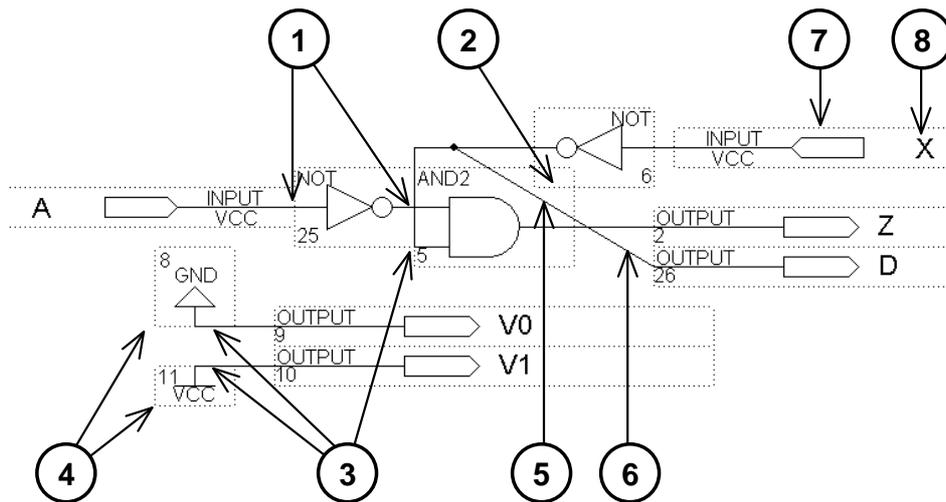
Auf den ersten Blick lesbare und korrekte Schemas zu zeichnen ist eine Kunst die alle einfach lernen können. Zugegeben, ein Schema einfach rasch-rasch und schludrig zu zeichnen geht schneller. Sobald jedoch Fehler gesucht oder Ergänzungen zugefügt werden müssen, ist gewonnene Zeit sofort wieder verloren.

Der MAX+plusII Compiler ist zwar sehr nachsichtig und interpretiert chaotische Schemas zu oft richtig, so wie es eigentlich gemeint war, - aber eben nicht immer, und dann kann man unverständlichen Fehlermeldungen gegenüberstehen und hat die vorher beim schludrigen zeichnen des Schemas gewonnene Zeit rasch 10-fach und mehr wieder verloren, - und notabene die Assistenten unnötig in Beschlag genommen.

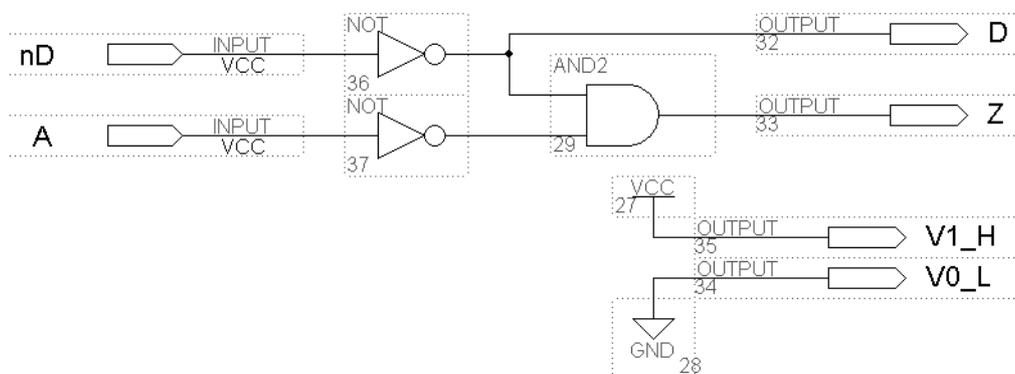
Die folgenden Regeln helfen ein Schema übersichtlich und vom Compiler fehlerlos lesbar zu gestalten:

- Signalfluss grundsätzlich von links nach rechts. Alle Eingänge links, alle Ausgänge rechts (7).
- Verbindungen nach Masse (GND) nach unten, nach Speisung (VCC) nach oben ziehen, falls notwendig horizontal aber nicht umgekehrt (4).
- Symbole anschlussseitig nicht direkt aneinander zeichnen (1), immer eine kurze Leitung dazwischen einfügen.
- Verbindungen zu Symbolanschlüssen immer vom Symbol weg, nie auf der Symbolkante ziehen (3).

- Symbole nie überlappen lassen (2). Das Aneinanderzeichnen ist erlaubt, wenn an den aneinander stossenden Seiten keine Anschlüsse sind.
- Nie eine Leitung über ein Symbol ziehen (5). Auf einer anschlussfreien Symbolkante ist gerade noch erlaubt.
- Leitungen nur horizontal und vertikal ziehen, schräge Leitungen vermeiden (6).
- Für nichts sagende Signalbezeichnungen Buchstaben vom Anfang des Alphabets für Eingänge, vom Ende (Z, Y, X, W...) für Ausgänge verwenden.



Schema schludrig in schlechtem Stil gezeichnet



Schema sorgfältig in gutem Stil gezeichnet