FPGAs

Familien Übersicht

Hier im Institut haben wir uns ziemlich auf die Altera Chips eingeschossen, aber es gibt auch andere Hersteller mit manchmal interessante Features.

- Xilinx
  - Spartan, Artix: low cost
  - Zynq: FPGA mit ARM
  - Virtex: high end. Unter andere in die NI-RIO Karten verbaut.
  - Ein par Virtex varianten gibt es auch als Rad-Hard.
- Altera
  - MAX: non-volatile, meist wenige LEs
  - MAX10: non-volatile wie MAX, fähig wie Cyclone, hat sogar analoge Eingängen (ADC).
  - Cyclone: Low cost
  - Arria: Mid range
  - Stratix: High end
  - Die letzte Modelle von Cyclone, Aria und Stratix haben auch ARM-Prozessoren dabei.
  - Ich meine das es auch noch so etwas wie einen Intel-Altera Combi-Chip gibt (x86 mit FPGA, ziemlich dickes Ding). Intel hat Altera auch aufgekauft.
- Lattice
  - ICE40 HX4K/HX8K kosten um 6 Euro, haben Logik (7680 LCs), Speicher (128 kbit), PLL und GPIOs, sind in einer ähnlichen Größenordnung wie die hier sonst verwendeten Altera MAX10, haben aber keine internen AD-Wandler.
  - ICE40 UP5K sind langsam, aber für sehr niedrigen Stromverbrauch optimiert.
  - ECP5 sind vergleichsweise große (bis 84K LCs) und teure Chips, die erst bei entsprechenden Anforderungen sinnvoll werden.
  - Lattice-FPGAs laden den Bitstream aus einem externen SPI-Flash und können auch über einen Microcontroller als SPI-Slave “gebootet” werden. Das Protokoll ist vollständig dokumentiert, ich habe bereits meinen eigenen Bootloader dafür geschrieben. Es werden keinerlei spezielle Tools gebraucht; die offiziellen Boards verwenden ein FTDI-USB-Seriell-Brücke zum Beschreiben des externen SPI-Flash. Für fertige Designs gibt es außerdem einen nur einmal beschreibbaren internen Konfigurationsspeicher.
  - Der ganz große Vorteil bei Lattice ist die Software: Es gibt eine Open Source Toolchain, die Verilog kompiliert und sich wie ein Kommandozeilencompiler für einen Microcontroller anfühlt. Wer mit Quartus/Vivado/Libero gekämpft hat, wird es zu schätzen wissen. https://symbiflow.github.io/ Project Icestorm für ICE40 http://www.clifford.at/icestorm/ und Project Trellis für ECP5.
  - HX4K und HX8K enthalten den gleichen Chip-Die, sind allerdings in verschiedenen Gehäusevarianten erhältlich: TQPF144 und verschiedene BGA. “HX4K” ist nur eine Beschränkung zur Marktsegmentierung in der Hersteller-Software “Diamond”.

    - Hinter der freien Toolchain steckt ein Projekt zur formellen Verifkation, die Software ist sehr stabil und ausgereift.
    - Alle besonderen Funktionsblöcke sind von Lattice aus offiziell dokumentiert, es ist nicht nötig, sich mit irgenwelchen Konfiguratoren herumzuschlagen.
    - Für eine Einführung: Dr. Matthias Koch, D229, fragen. Ich habe diese FPGAs in meiner Doktorarbeit verwendet und zwei verschiedene Prozessorkerne darauf implementiert, einer davon eine Eigenentwicklung. http://mecrisp.sourceforge.net/

- MicroSemi / Actel
    - Fusion: FPGA + Analog (einen ADC mit viele Features und deutlich höhere Spannungen als üblich)
    - SmartFusion: ARM-Prozessor + Analog (Input und höhere Spannungen) + FPGA
    - Igloo, ProAsic : basic low power FPGA
    - RT-xxx : Strahlungsharte FPGAs, von klein bis riesig. Nur die RTG4 hat PLLs und Multiplikatoren.
    - Multiplikatoren gibt es nur in wenige Baureihen.

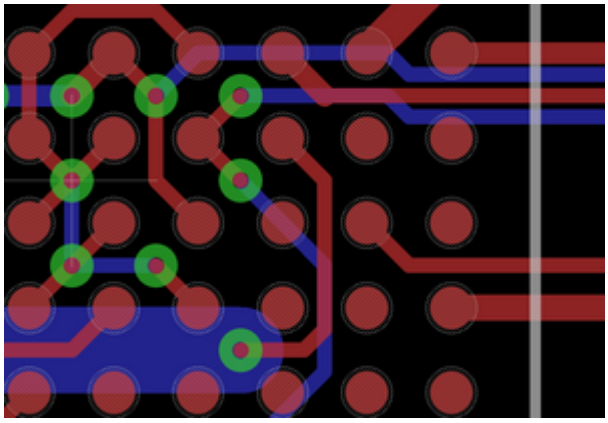
- (Diese Liste erhebt kein Anspruch auf Vollständigkeit. Es sind nur die übliche Verdächtige.)

Vorschläge für Verdrahtung

Jede größere FPGA gibt es nur noch in BGA-Gehäusen (Ball Grid Array). Wie kommt man an die Pins ran? Mit der Lötkolbe geht da nichts mehr. Dies geht nur noch mit Reflow-Löten im Ofen oder noch besser mit Dampflöten. Am besten geht es aber wenn man dies an Bestückungsfirmen auslagert.

Wir müssen dann nur noch den Entwurf machen und die richtige Technik-Optionen auswählen.

- Die TQFP Gehäusen sind noch mit Standart Einstellungen in 8mil-Technik realisierbar.
- Für FBGAs mit 1.0mm Raster gibt es eine Lösung die mit einfache Zusatzoptionen realiserbar ist. Diese Gehäuse machen aber erst Sinn wenn man mindestens einen 4-Lagen Board entwerft.
    - Die äußere beide Reihen kann man direkt erreichen. Bei einer Pad-Durchmesser von 0.5mm, braucht man 7mil Linienbreite und 6mil Isolation. Beides ist mit normaler 6mil-Technik möglich.
    - Für die innere Pins ab der dritte Reihe sind Vias nötig.
        - Drill: 8mil, Restring: 6mil, Durchmesser: 20mil (8+2*6).
        - Dies sind meistens Sonderoptionen.
    - Die kleine Vias sollen mit Stoplack überdeckt sein (Eagle: DRC $\longrightarrow$ Masks $\longrightarrow$ Limit=0.4mm). Dies verhindert Kurzschlüsse unter dem Chip.
    - Die Öffnungen im Stoplack für die Pads sollten nur 3mil größer sein als die Pads (Eagle: DRC $\longrightarrow$ Masks $\longrightarrow$ Stop-Min=Stop-Max=3mil).
    - Da diese Vias ziemlich klein sind, sollte man vorallen für Ground und Versorgung viele Vias nutzen.

- Ein Vorbild für wie es dann aussieht: 
- Für UBGA und MBGA hab ich noch keine Lösung gefunden. Man braucht hier meistens Spezial-Optionen aus er Kategorie “auf Anfrage”.
  - Diese Gehäusen sind für uns zu klein. Aber sag niemals nie. Spätestens wenn irgendjemand ein Atominterferometer im Händy haben will, werden wir auch diese Chips einsätzen.
  - MBGAs sind nur noch mit Via-in-Pad möglich.
- Weitere Links zu genaue Platinen-Maßen:
  - https://www.multi-circuit-boards.eu/leiterplatten-design-hilfe/bga-leiterplatte-ball-grid-array.html
  - http://www.xilinx.com/support/documentation/user_guides/ug1099-bga-device-design-rules.pdf
  - https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an114.pdf

Zum selberlöten von BGAs gab es auch mal eigene Versuche auf eine andere Seite im Wiki: Löten von Ball Grid Array

Altera

Altera Gehäusen

Code	Kurzname	Volle Name	Rasterabstand	Anmerkung
T100	TQFP100	Thin Quad Flat Pack 100 pins	0.5mm	
E64	EQFP64	Enhanced Quad Flat Pack 64 pins	0.4mm	Sieht aus wie TQFP64 ist aber viel kleiner, mit Pad auf Unterseite
E144	EQFP144	Enhanced Quand Flat Pack 144 pins	0.5mm	Dies ist eine TQFP144 mit extra Pad auf Unterseite
F256	FBGA256	Fineline Ball Grid Array 256 pins	1.0mm	
U256	UBGA256	Ultra Fineline Ball Grid Array 256 pins	0.8mm	
M256	MBGA256	Micro Ball Grid Array 256 pins	0.5mm	

- Die Zahl hinter der Buchstabe ist variabel.
- Der Spaß bei E64 ist ein Beispiel dafür das man trotz vermeintlich bekanntes Aussehen doch noch mal im Datenblatt nach der Zeichnung schauen soll (ein TQFP64 hat normalerweise

0.8mm Raster). Und wer sucht findet noch viel mehr TQFP Gehäusen mit unterschiedliche Raster.

Manchmal werden auch andere Codes für die Gehäuse verwendet, diesmal benannt nach der Länge des Gehäuse in millimeter:

Längencode	Pincode	Pins	Raster
F17	F256	16 x 16	1.0mm
F23	F484	22 x 22	1.0mm
F27	F627	26 x 26	1.0mm
F31	F896	30 x 30	1.0mm
U15	U324	18 x 18	0.8mm
U19	U484	22 x 22	0.8mm
M13	M383	18 x 18	0.5mm
M15	M484	22 x 22	0.5mm

Altera Pinbelegung

Bei den QFP Gehäusen ist die Pinbelegung kein wirklich wichtiges Auswahlkriterium. Versorgung, Ground und IO liegen immer gemischt aber gut erreichbar.

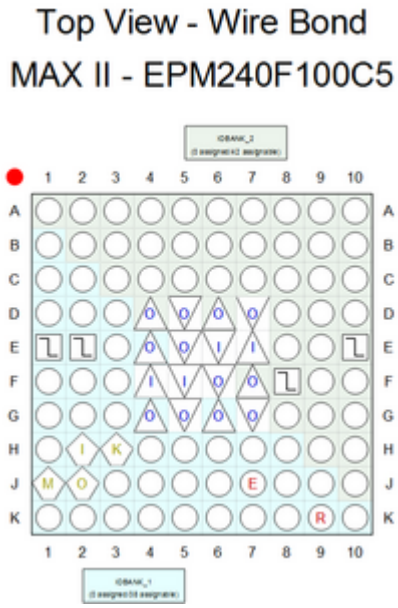
Bei den BGAs sieht das anders aus. Dort sind die innere Pins schwierig zu routen, zumindest wenn man nur 4 Lagen hat und nur durchgehende Vias durch alle Lagen. Man möchte deshalb möglichst viele Signalpins auf den äußere beide Reihen haben. Manche der Pinbelegungen sind in dieser Hinsicht sub-optimal.

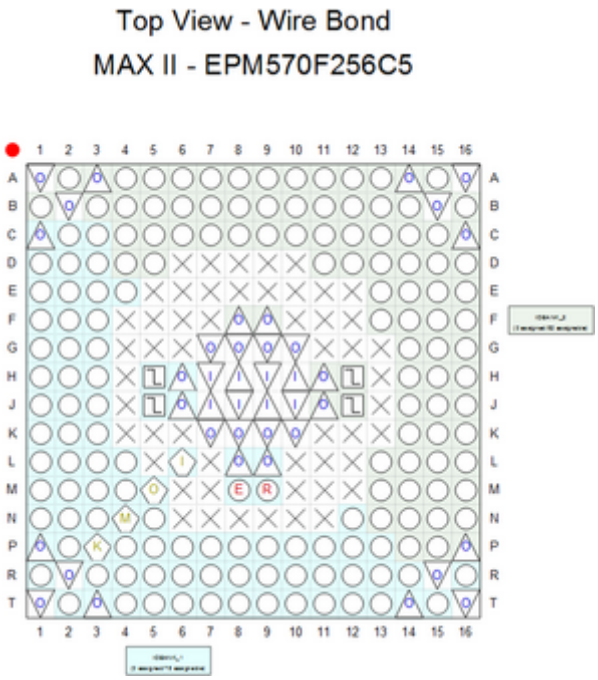
Die nachfolgende Bilder kommen alle aus der Pin-Planner von Quartus.

Kurze Symbolerklärung:

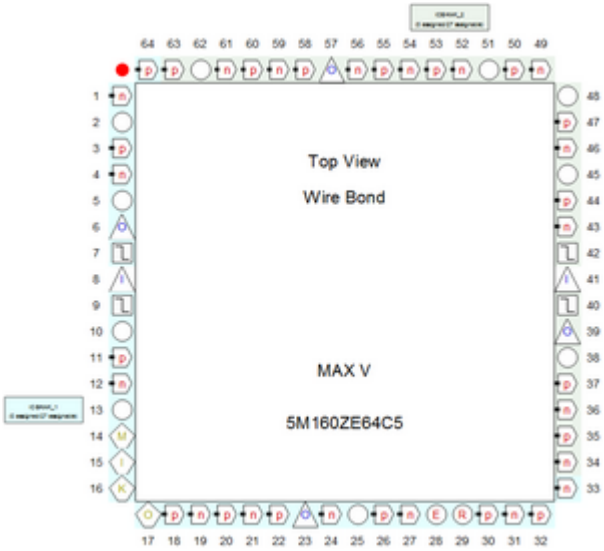
- Dreieck nach oben: Versorgung. Das Zeichen in dem Dreieck sagt welche Versorgung.
- Dreieck nach unten: Ground
- Pentagram: Programmierung
- Vierkant mit steigende oder fallende Flanke: Clock
- Rund oder Hexagon: IO
- Kreuz: Nicht angeschlossen
- Farbe: zeigt die jeweile IO-Bank an

MAX2



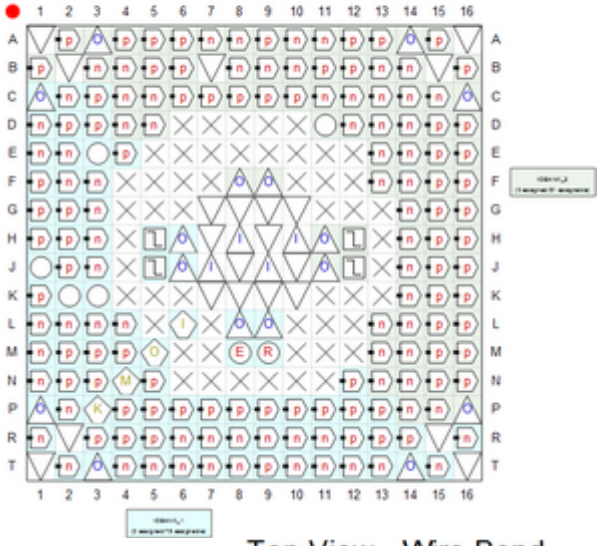


MAX5

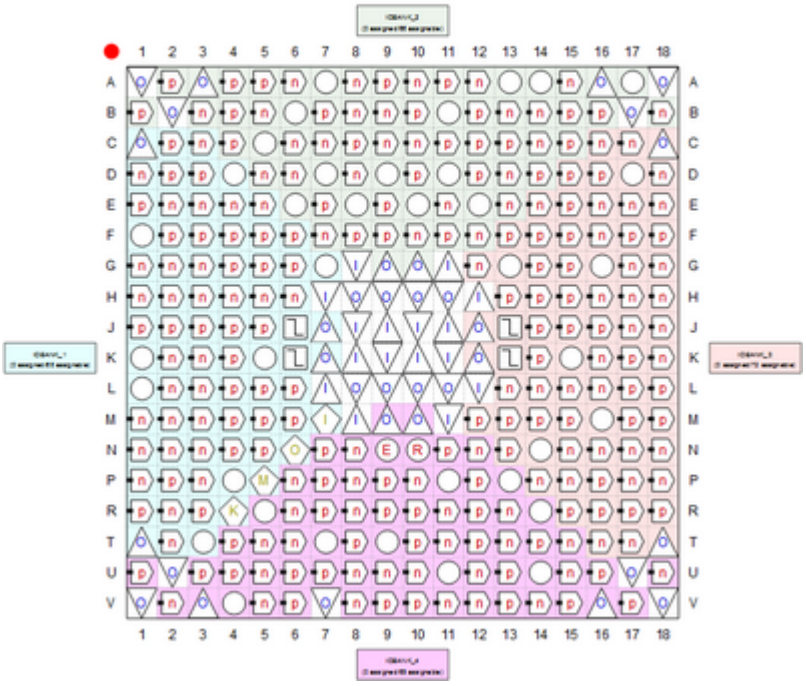


Bei der MAX5 in E64 Gehäuse fehlen seldsamerweise sämtliche Ground Pins. Ground ist hier auf der Unterseite.

Top View - Wire Bond
MAX V - 5M570ZF256C5



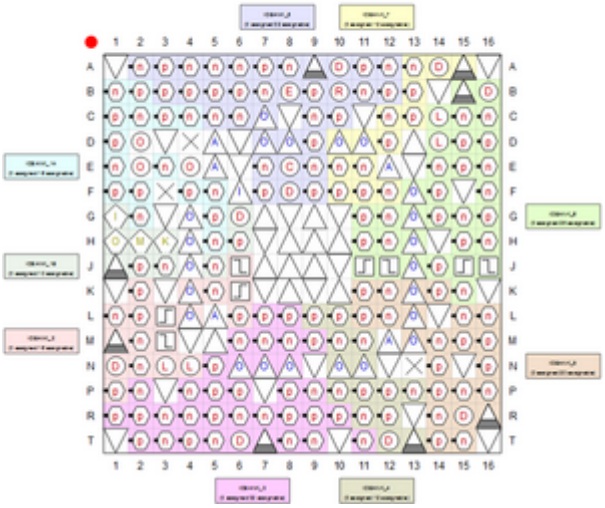
Top View - Wire Bond
MAX V - 5M1270ZF324C5



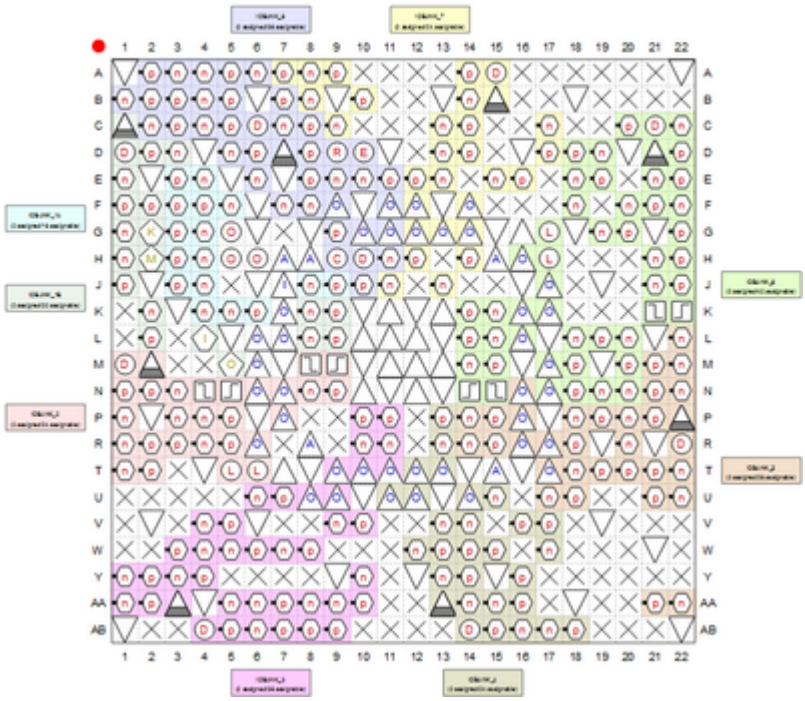
Es gibt eine auffällige Übereinstimming zwischen den MAX2 und MAX5 Chips im gleichen Gehäuse.

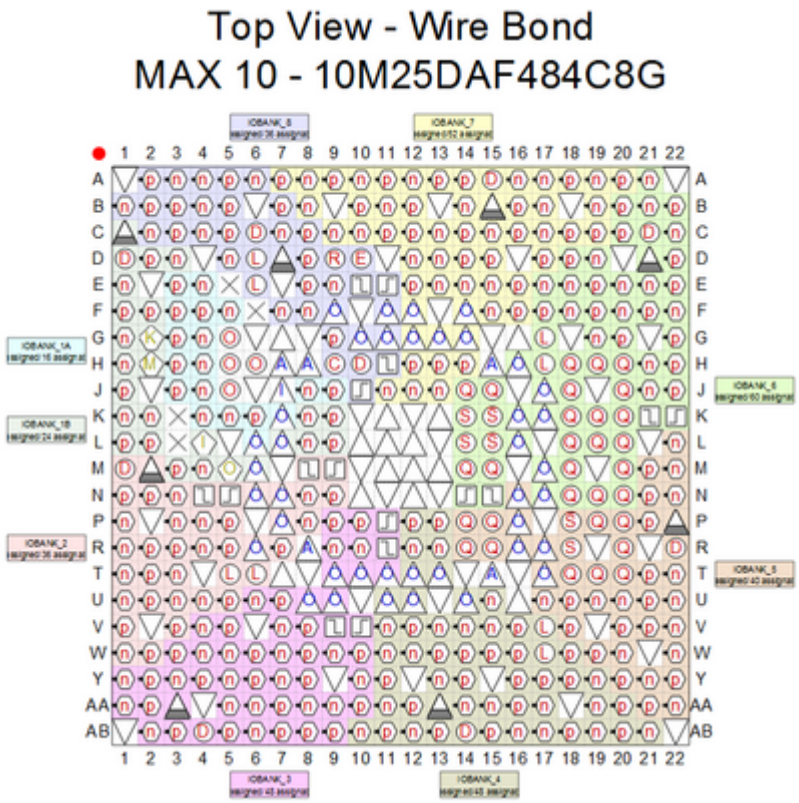
MAX10

Top View - Wire Bond
MAX 10 - 10M08DAF256C8G



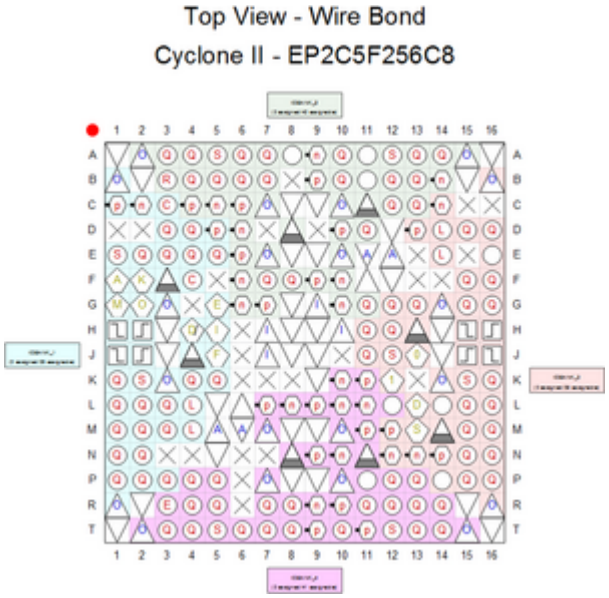
Top View - Wire Bond
MAX 10 - 10M08DAF484C8G

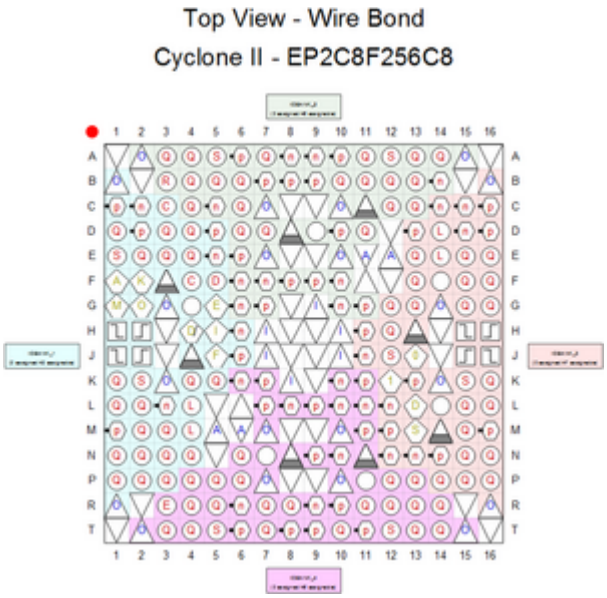




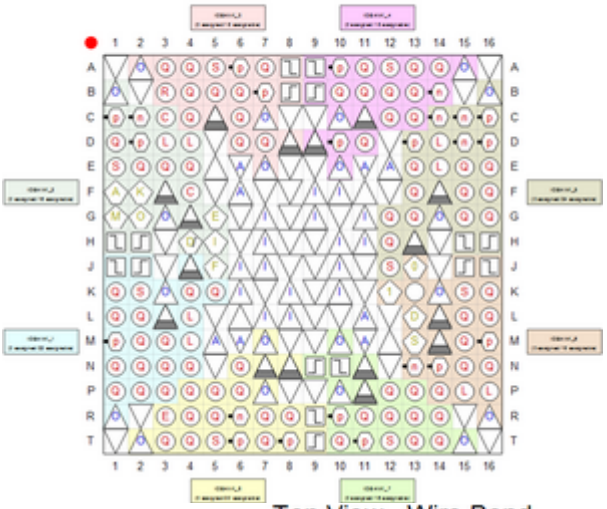
Die Pinbelegung der andere MAX10-Varianten unterscheidet sich im gleichen Gehäuse nicht. Zumindest in die meiste Fälle, sieh Datenblatt. Nur ein par Pins ändern sich wenn man zwischen der Analog und der Compact-Version wechselt.

Cyclone 2

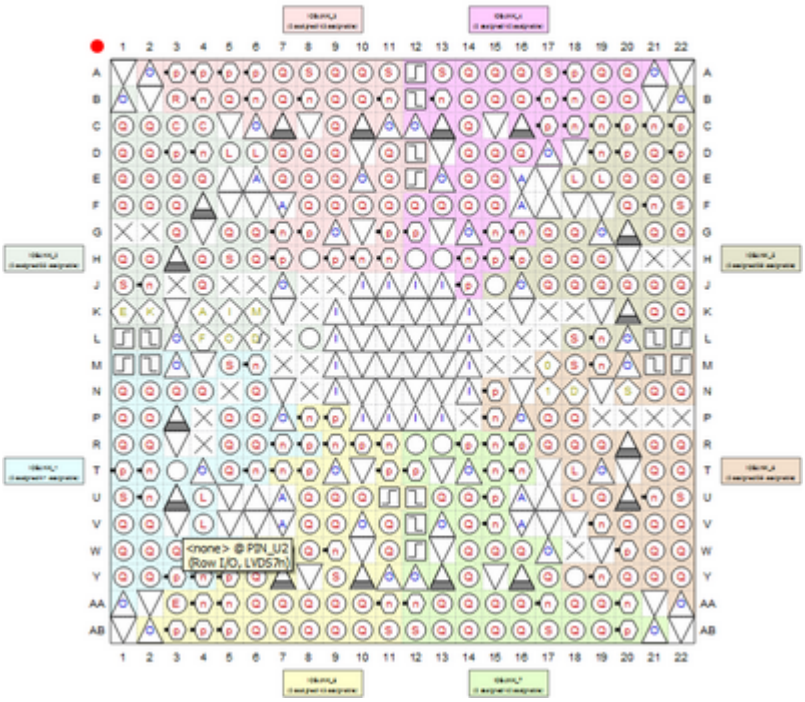




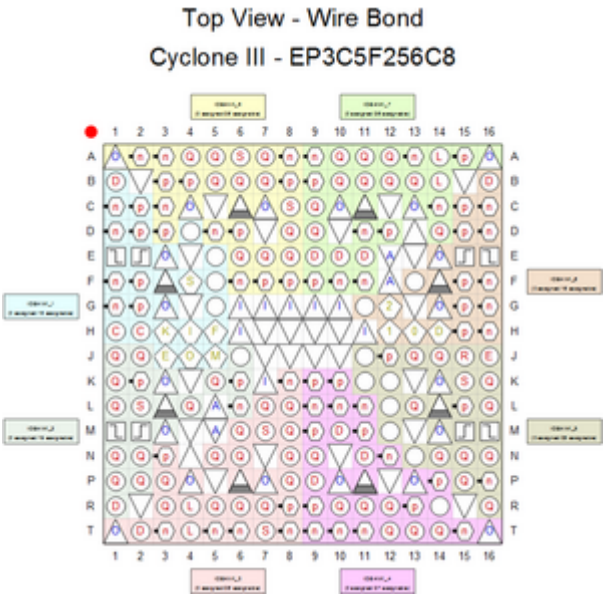
Top View - Wire Bond
Cyclone II - EP2C20F256C8



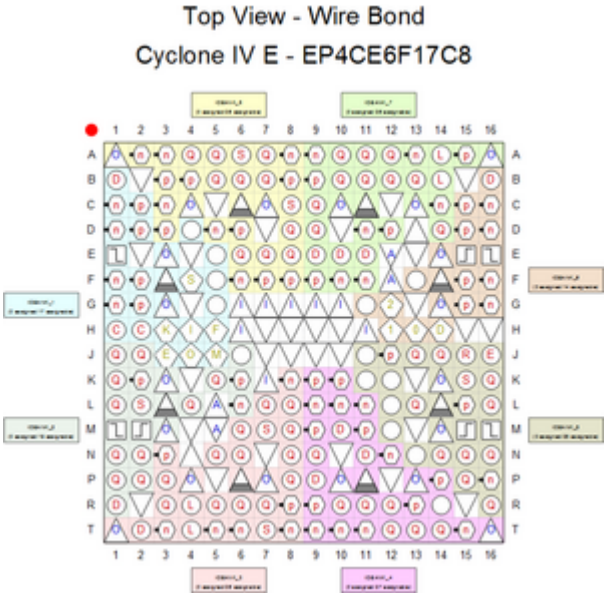
Top View - Wire Bond
Cyclone II - EP2C20F484C8

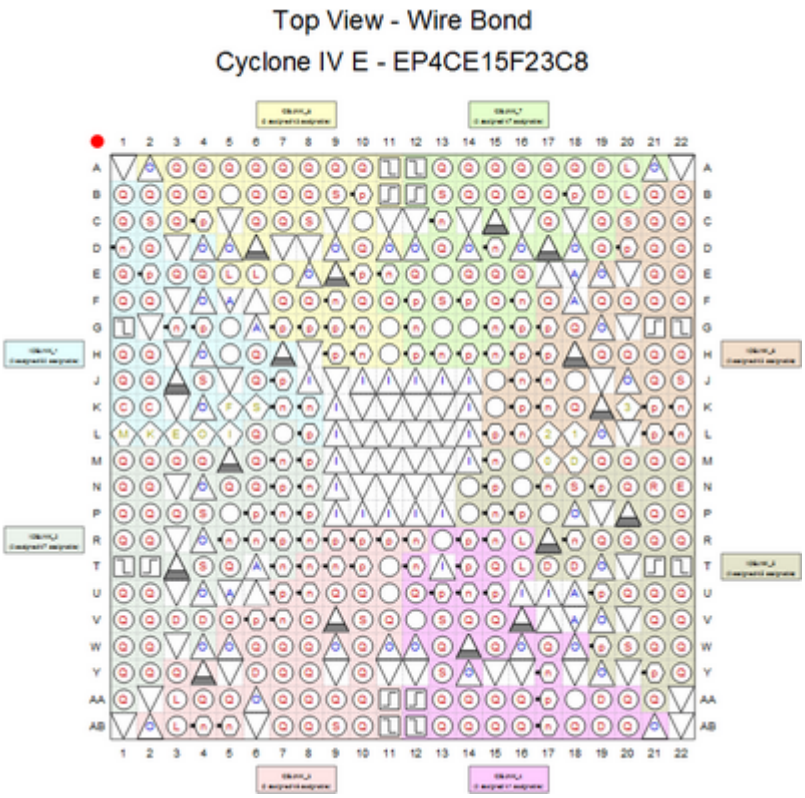


Cyclone 3

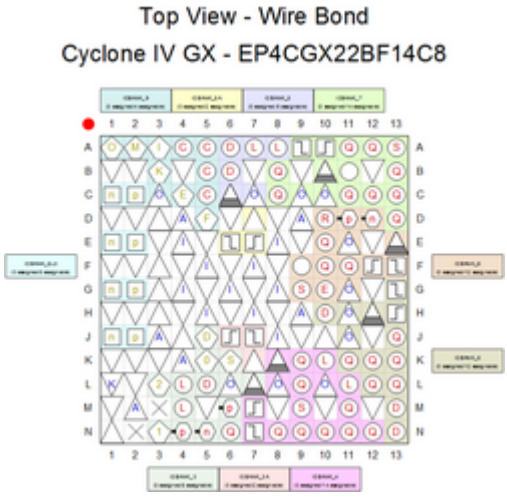

Top View - Wire Bond
Cyclone III - EP3C5F256C8

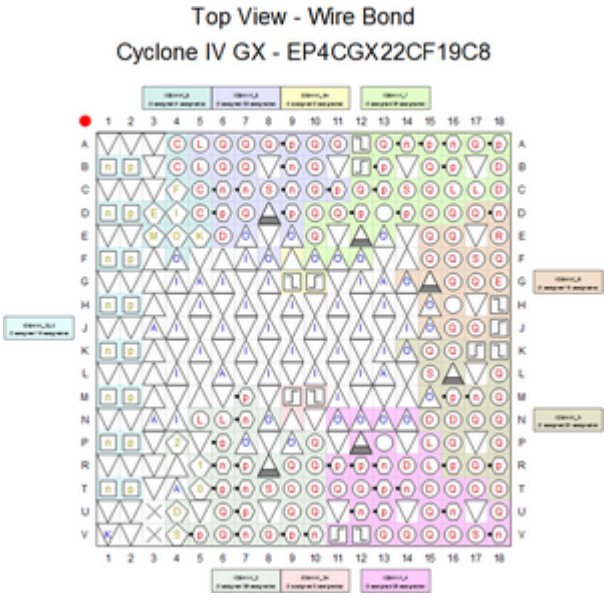
Cyclone 4

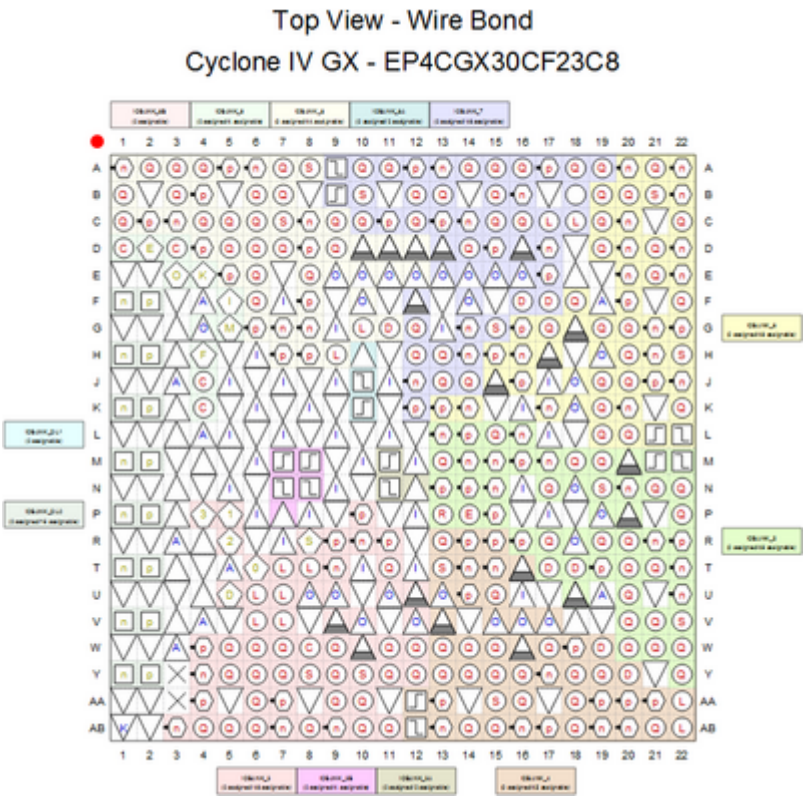

Top View - Wire Bond
Cyclone IV E - EP4CE6F17C8


Top View - Wire Bond
Cyclone IV E - EP4CE15F23C8

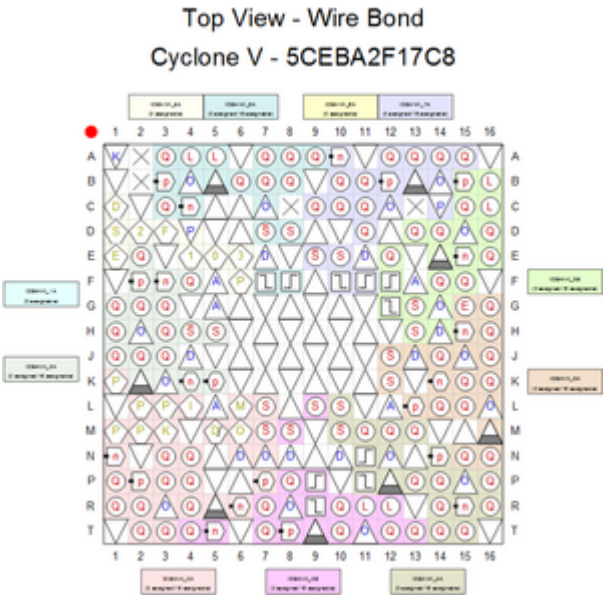
Cyclone 4GX


Top View - Wire Bond
Cyclone IV GX - EP4CGX22BF14C8

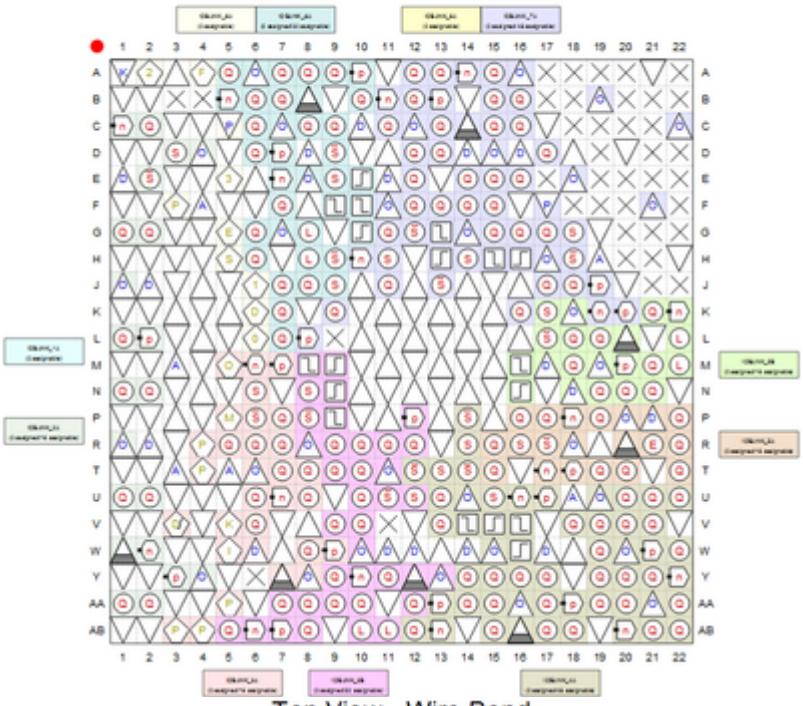

Top View - Wire Bond
Cyclone IV GX - EP4CGX22CF19C8


Top View - Wire Bond
Cyclone IV GX - EP4CGX30CF23C8

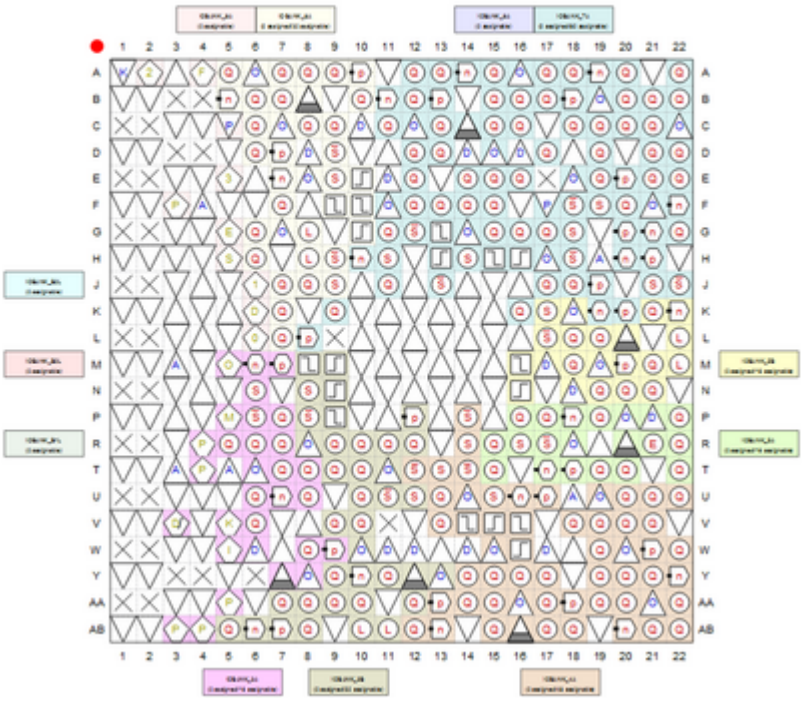
Cyclone 5E


Top View - Wire Bond
Cyclone V - 5CEBA2F17C8

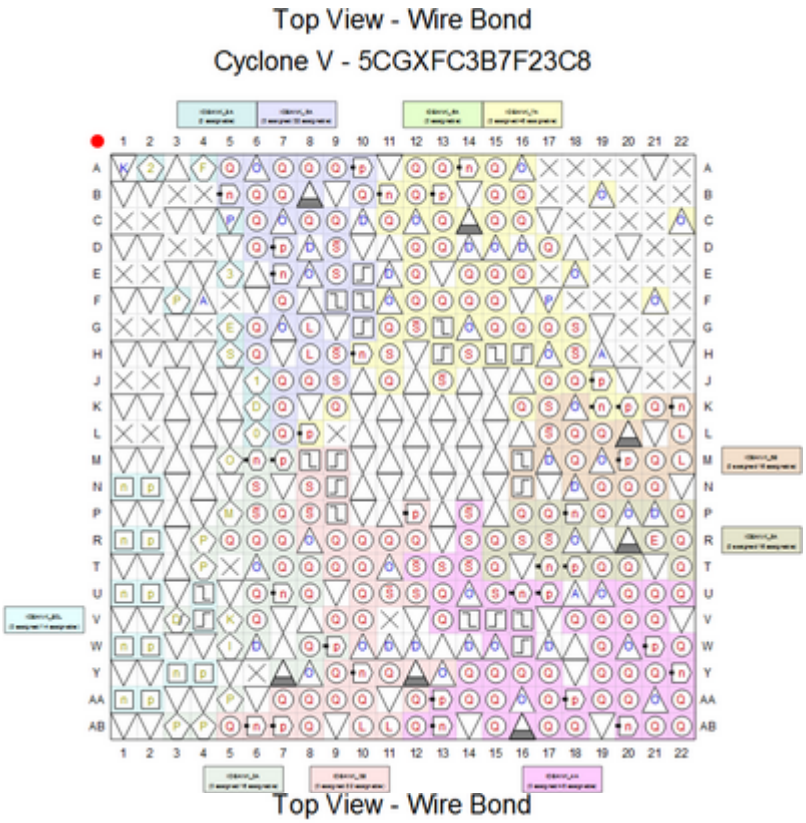
Top View - Wire Bond
Cyclone V - 5CEBA2F23C8

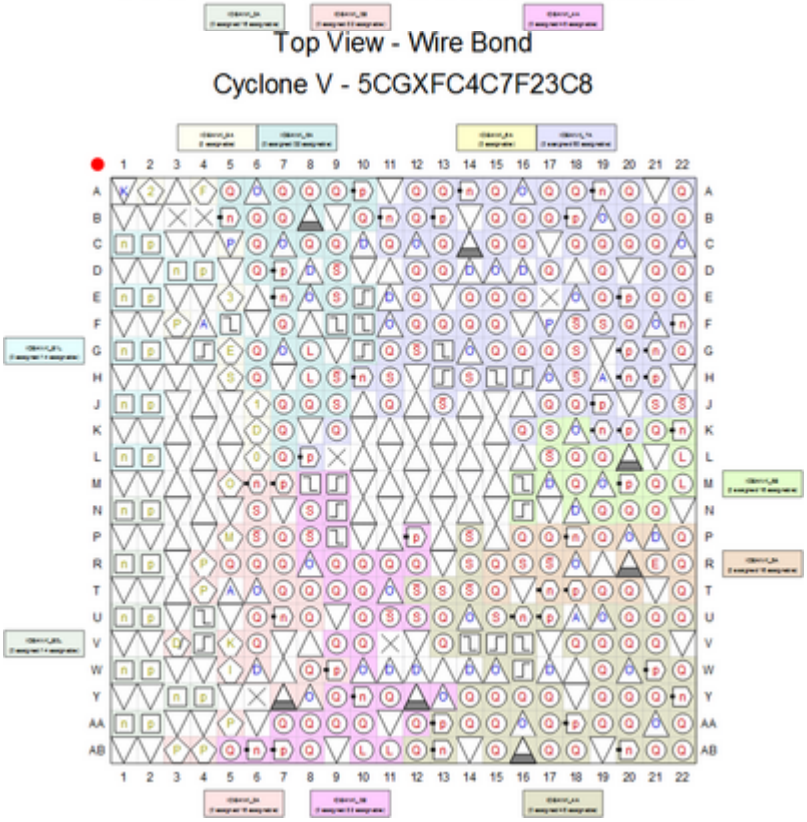


Top View - Wire Bond
Cyclone V - 5CEBA7F23C8



Cyclone 5GX





Altera Quartus Versionen

(Wir betrachten hier nur für die Gratis-Version.)

Einen Upgrade bringt hier leider nicht nur neue Sachen und Verbesserungen, man kann bei Quartus auch alte Features verlieren. Zum Glück gibt es eine friedliche Koexistenz der Versionen. Man kann sie ohne Stress neben ein ander betreiben. Und man kann die alte Versionen noch Downloaden.

Stress gibt es nur wenn man bestehende Projekte mit der falsche Version öffnet. Dabei will Quartus

bereits beim Öffnen, Projektdateien überschreiben, wofür er auch warnt.

Version	Simulator	MAX7000	MAX3000	MAX2	MAX5	MAX10	Cyclone1	Cyclone 2	Cyclone 3	Cyclone 4	Cyclone 5
9	Build-in, Modelsim	v	v	v			v	v	v		
10	Modelsim	v	v	v	v		v	v	v	v	
11	Modelsim	v	v	v	v		v	v	v	v	v
12	Modelsim	v	v	v	v			v	v	v	v
13.0	Modelsim	v	v	v	v			v	v	v	v
13.1	Modelsim			v	v				v	v	v
14	Modelsim			v	v	v				v	v
15	Modelsim			v	v	v				v	v
16	Modelsim			v	v	v				v	v

- Der einfach zu benutzende eingebaute Simulator gibt es nur bis Version 9 (und vielleicht noch als Academic Sonder-Download).
- Für MAX3000, MAX7000, Cyclone2 und Cyclone3 gilt: nur bis Version 13.0.
- Für alle neuere gilt: nimm die neueste Quartus Version (16).

From:
https://elektroniq.iqo.uni-hannover.de/ - **ElektronIQ**

Permanent link:
https://elektroniq.iqo.uni-hannover.de/doku.php?id=bauteil:fpgas

Last update: **2019/10/30 11:31**