PALs, CPLDs und FPGAs

Bezeichnungen

- Sehr ähnliche Bauelemente werden oft unterschiedlich bezeichnet, z.T. nur aus Marketing-Gründen

- PLD = Programmable Logic Devices

- PLA = Programmable Logic Array
- PAL = Programmable Array Logic
- CPLD = Complex Programmable Logic Devices
- FPGA = Field Programmable Gate Array (eigentlich sind das gar keine Gate Arrays...)
- LCA = Logic Cell Array

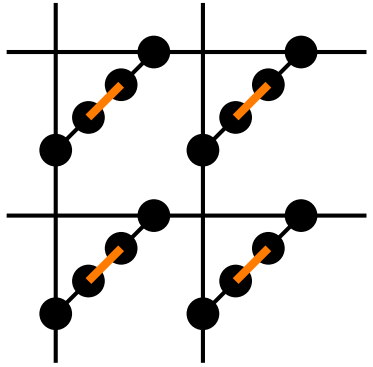
Firmen

- Der Markt für 'kleine' PALs geht zurück – nur noch wenige Firmen stellen PALs her
- Bei den 'großen' Bausteinen findet z.Z. eine **Konzentration auf wenige Firmen** statt.
- Marktanteile und Umsatz 2002 (Quelle: iSuppli) von PLD Herstellern:

Hersteller	Umsatz (Mio. $)	Marktanteil
Xilinx	1124	49.2 %
Altera	712	31.2 %
Lattice	229	10.0 %
Actel	134	5.9 %
Cypress	44	1.9 %
QuickLogic	31	1.4 %
Atmel	11	0.5 %
...	...	...

Programmiermethoden

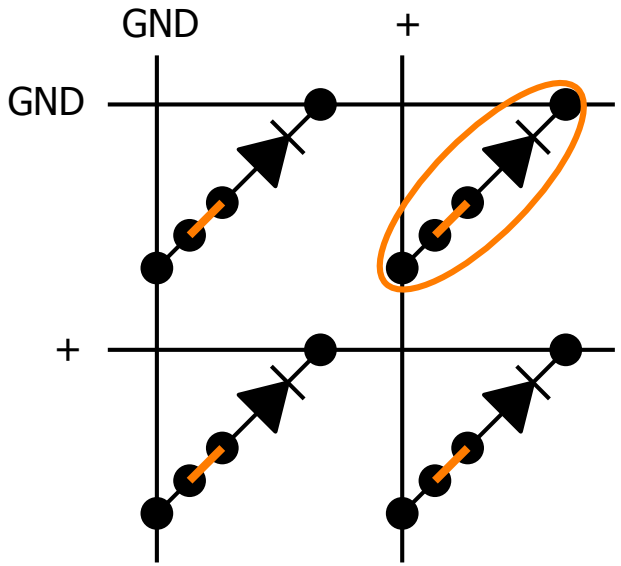
- Einmal programmierbar (one time programmable, **OTP**)
  - 'Fuse' (Sicherung) oder Anti-fuse Varianten. Verbindung wird hergestellt oder durchgebrannt
  - typ. Widerstände von 50-300 Ohm (niedrig)
  - sehr hohe Dichte: ein Bit benötigt wenig Platz
  - Programmierung erfolgt extern mit spezieller Hardware (‚PROMMER')



Durchbrennen einzelner Kreuzungen nicht möglich

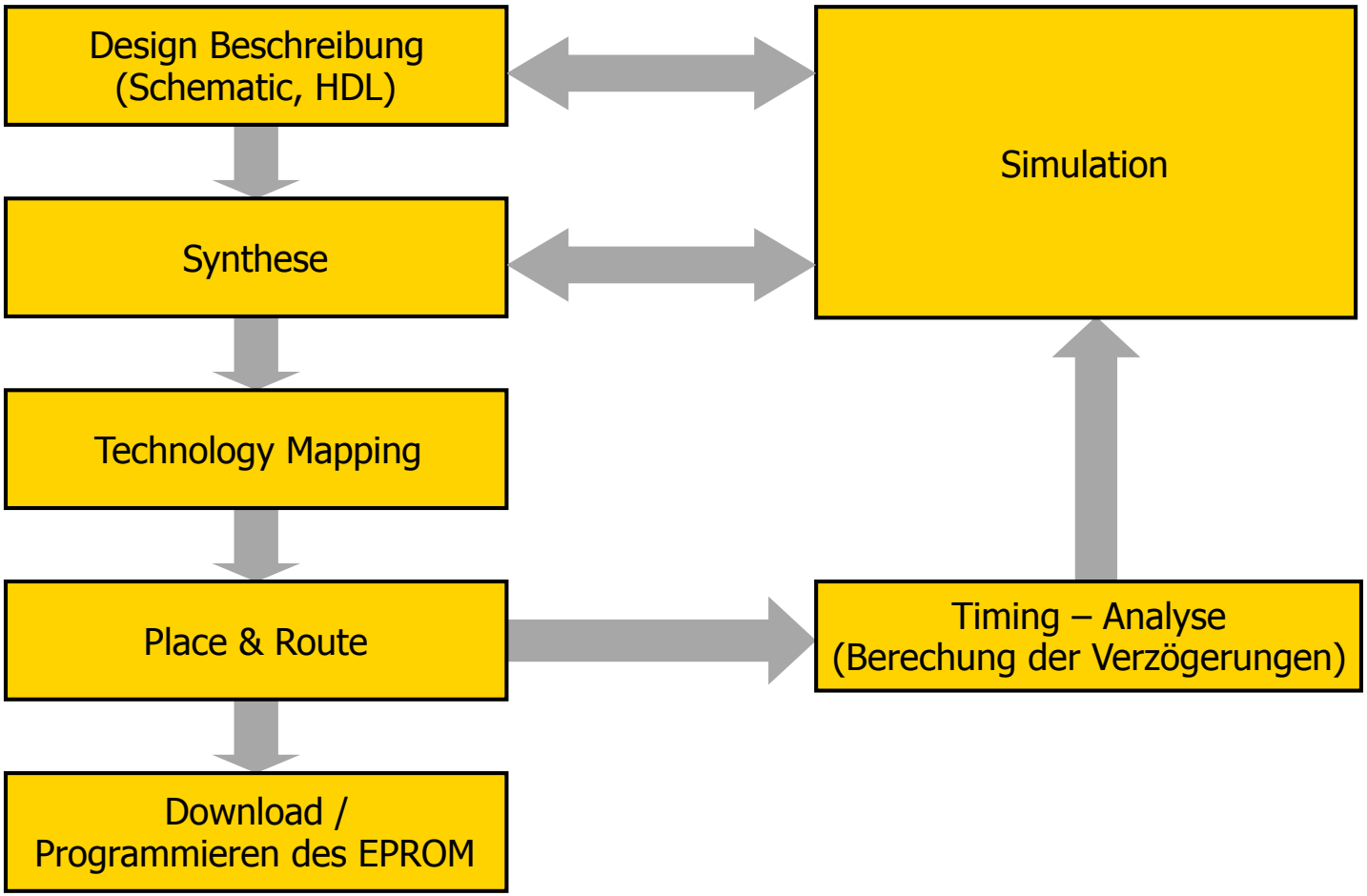
- **RAM**-basiert
  - RAM Bits kontrollieren CMOS Schalter (in unterschiedlichen Topologien)
  - Widerstände 0.5-1 kOhm
  - kann beliebig oft programmiert werden
  - kann auf der Platine programmiert werden: 'in circuit programmable'
  - niedrige Dichte
  - erfordert externen Speicher für Firmware, Chip muß 'booten'

- **EPROM**, **EEPROM und Flash** EEPROM
  - eigentlich ‚Read Only Memory' (ROM), diese Typen sind aber löschbar:
  - EPROM: (‚Erasable PROM') Löschen mit Licht
  - EEPROM: (‚Electrically Erasable PROM') Löschen elektrisch
  - Flash: Wie EEPROM, nur wesentlich schneller
  - kein externer Speicher nötig
  - Bauteil sofort nach Einschalten betriebsbereit: 'instant-On'
  - typ. Widerstände 2-4 kOhm
  - relativ hohe Dichte
  - spezielle Technologie nötig



Mit Dioden: Stromfuß nur in einer Kreuzung!

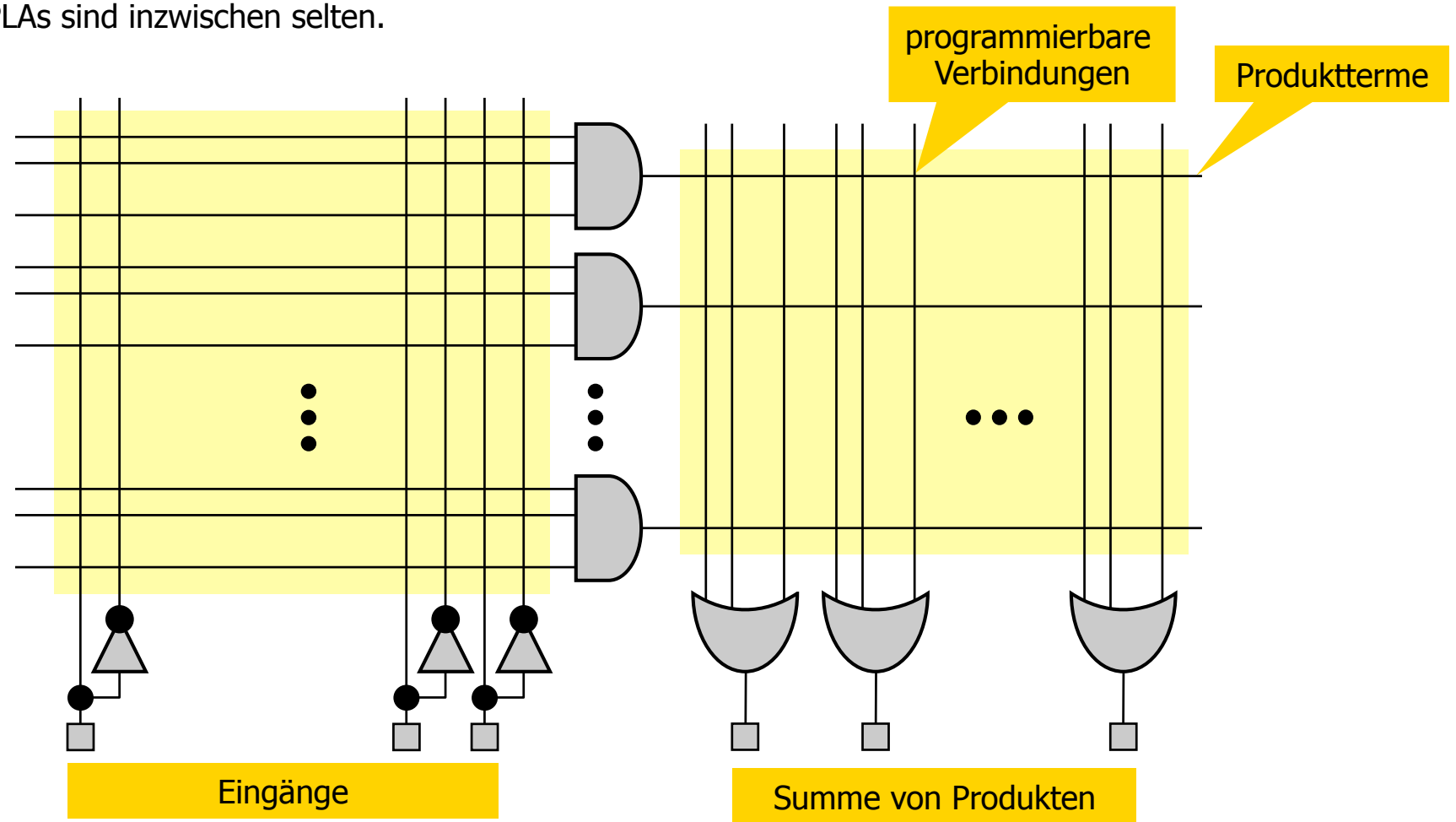
'Design Flow'



PLA, PROM, PAL

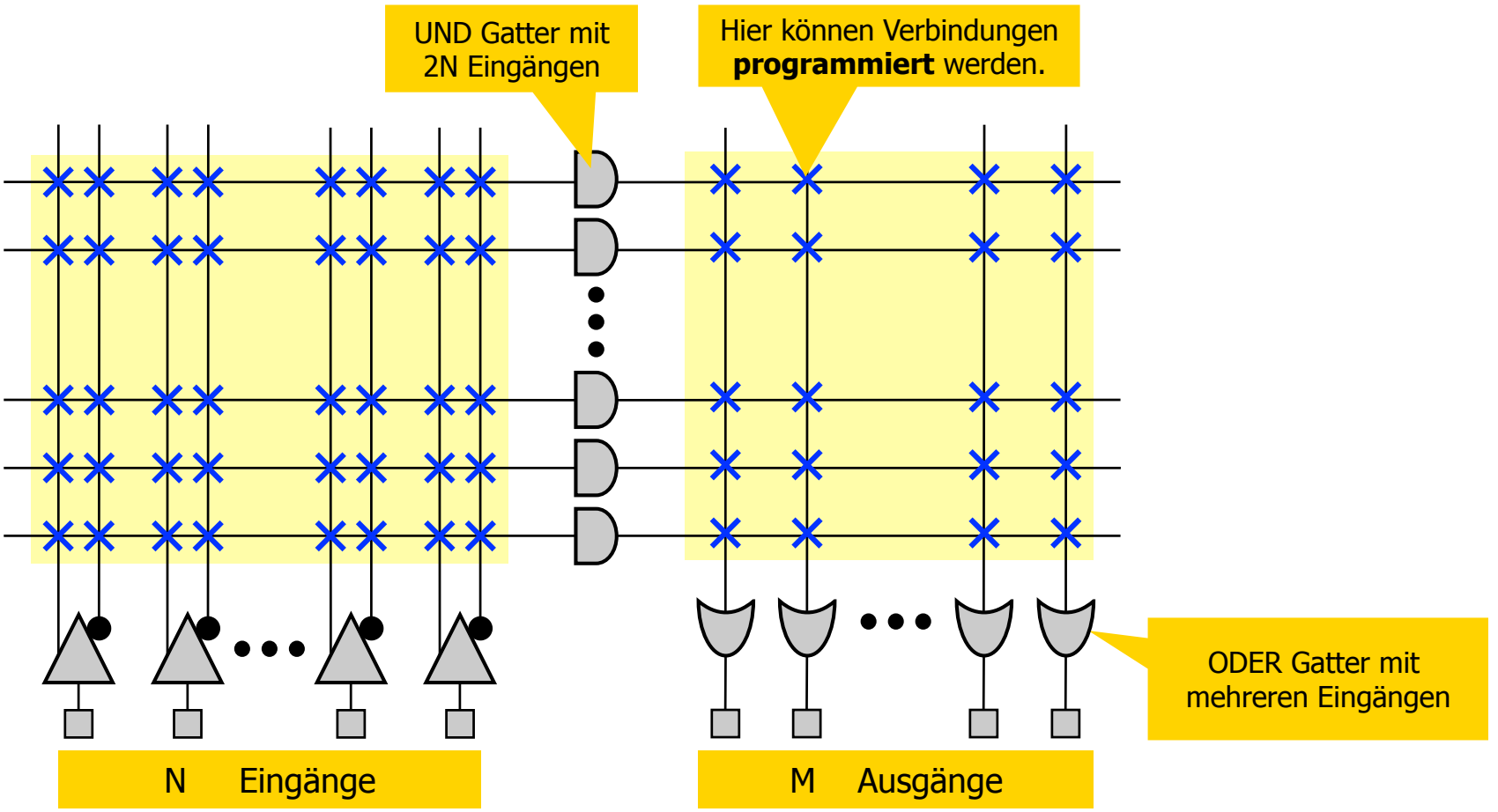
PLA

- PLAs enthalten eine programmierbare UND-Matrix, gefolgt von einer programmierbaren ODER-Matrix
- Zu jedem UND Gatter werden die Eingangssignale und deren Komplemente geführt
- Jeder Produktterm kann von mehreren ODER Gattern benutzt werden
- PLAs sind langsamer als PALs (s. später) durch die zwei programmierbaren Ebenen
- Reine PLAs sind inzwischen selten.



PLA: kompakte Darstellung

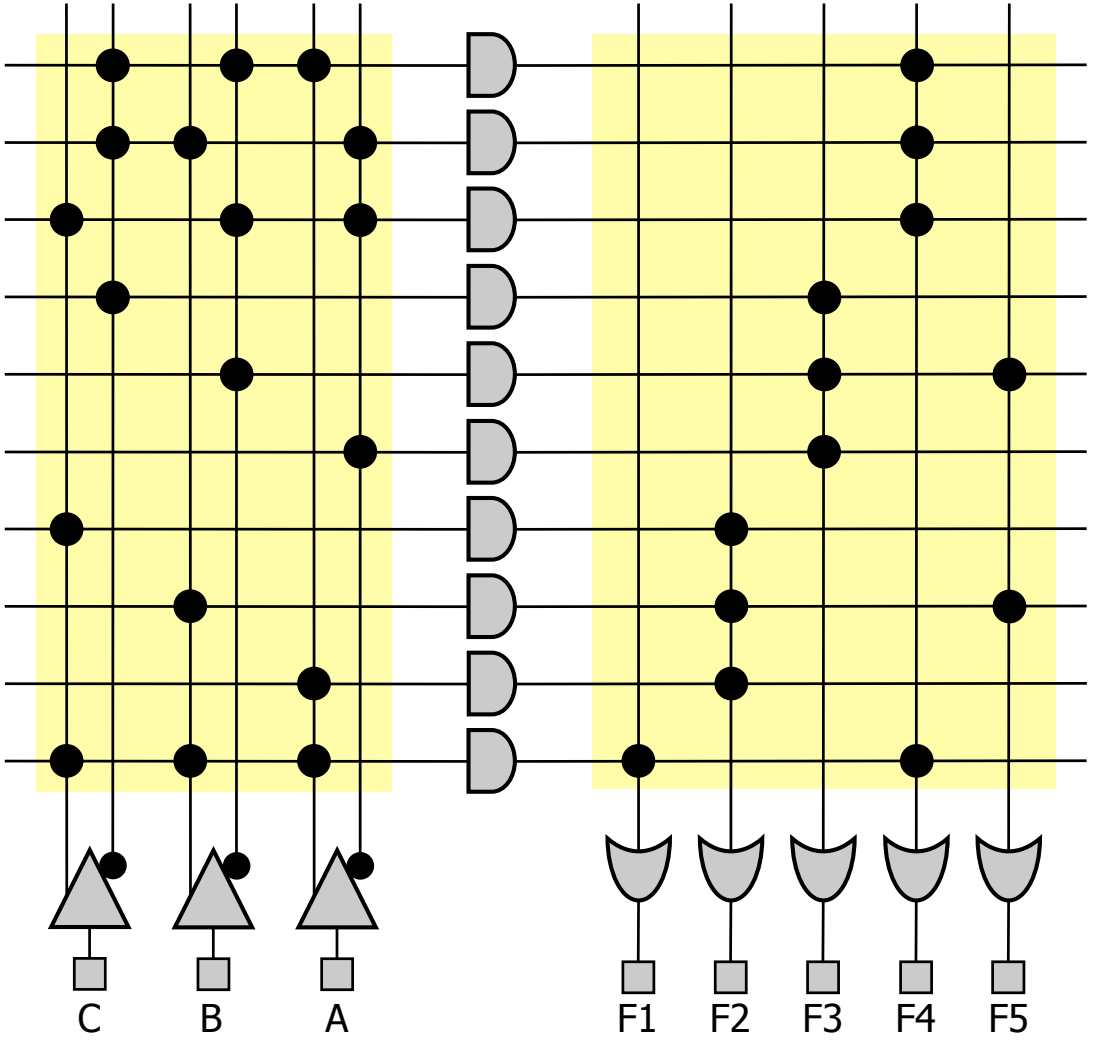
- Zur einfacheren Darstellung zeichnet man nicht alle Eingänge der Gatter
- Die programmierbaren Kreuzungen werden mit einem ‚x' markiert



Beispiele für Funktionen im PLA

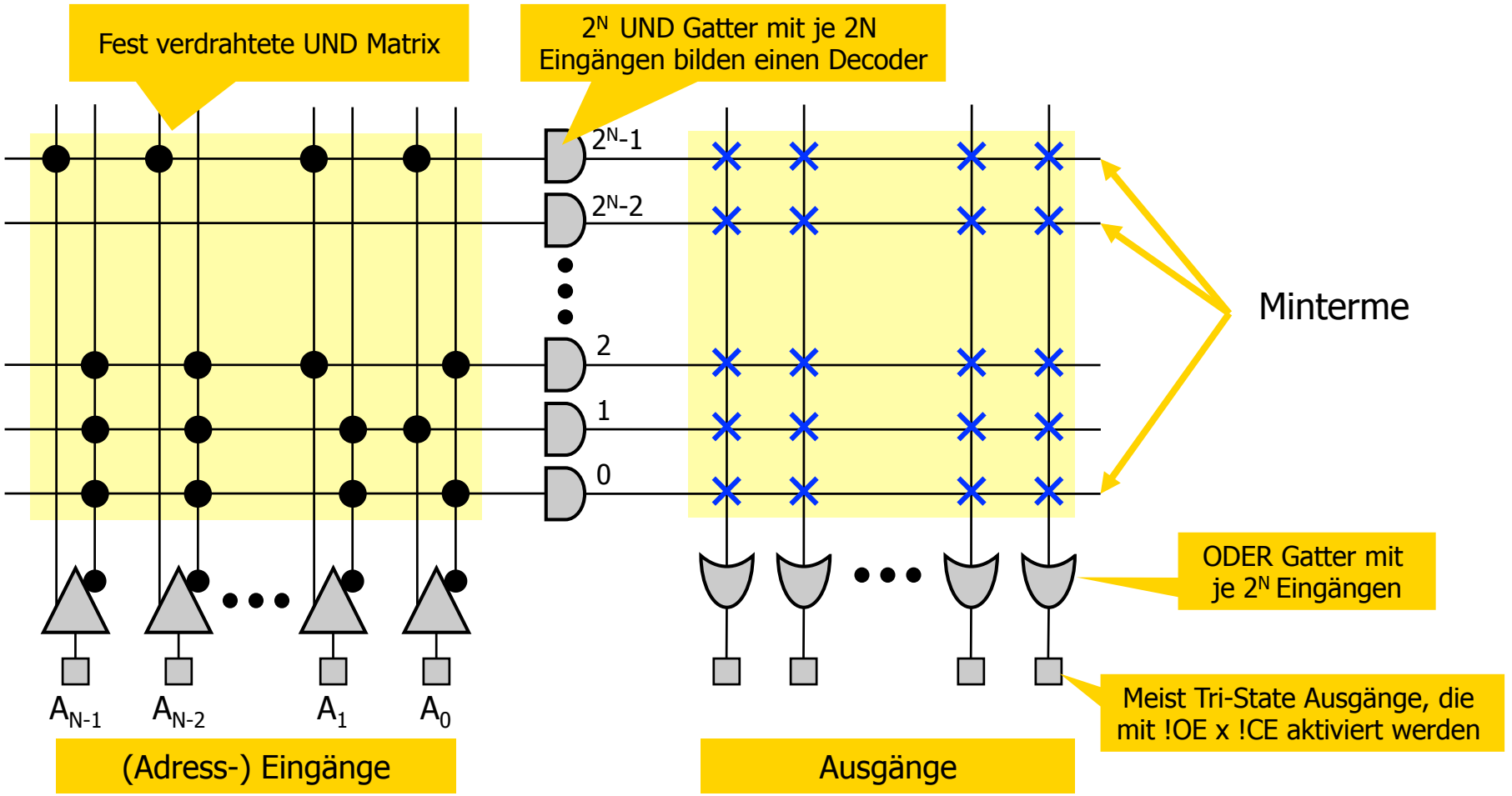
- F1 = ABC
- F2 = A + B + C
- F3 = !(ABC) = !A + !B + !C
- F4 = A ⊕ B ⊕ C
  = (AB + !A!B) C + (!AB + A!B) !C
  = ABC + !A!BC + !AB!C + A!B!C
- F5 = 1
  = B + !B

- Bei F4 und F5 werden Zeilen verwendet, die schon bei F1-F3 vorkommen
- Nicht jede Funktion kann implementiert werden, solange die Anzahl der zur Verfügung stehenden Produktterme begrenzt ist !



PROM – Wie es manchmal eingeführt wird...

- Im Vergleich zum PLA:
  - Die UND Struktur ist fest verdrahtet. Bei N Eingängen gibt es 2^N UND Gatter (Sie bilden einen Dekoder!)
  - Die ODER-Struktur legt die Ausgangswerte fest
- Jedes beliebige Muster ist möglich, aber sehr aufwändig!

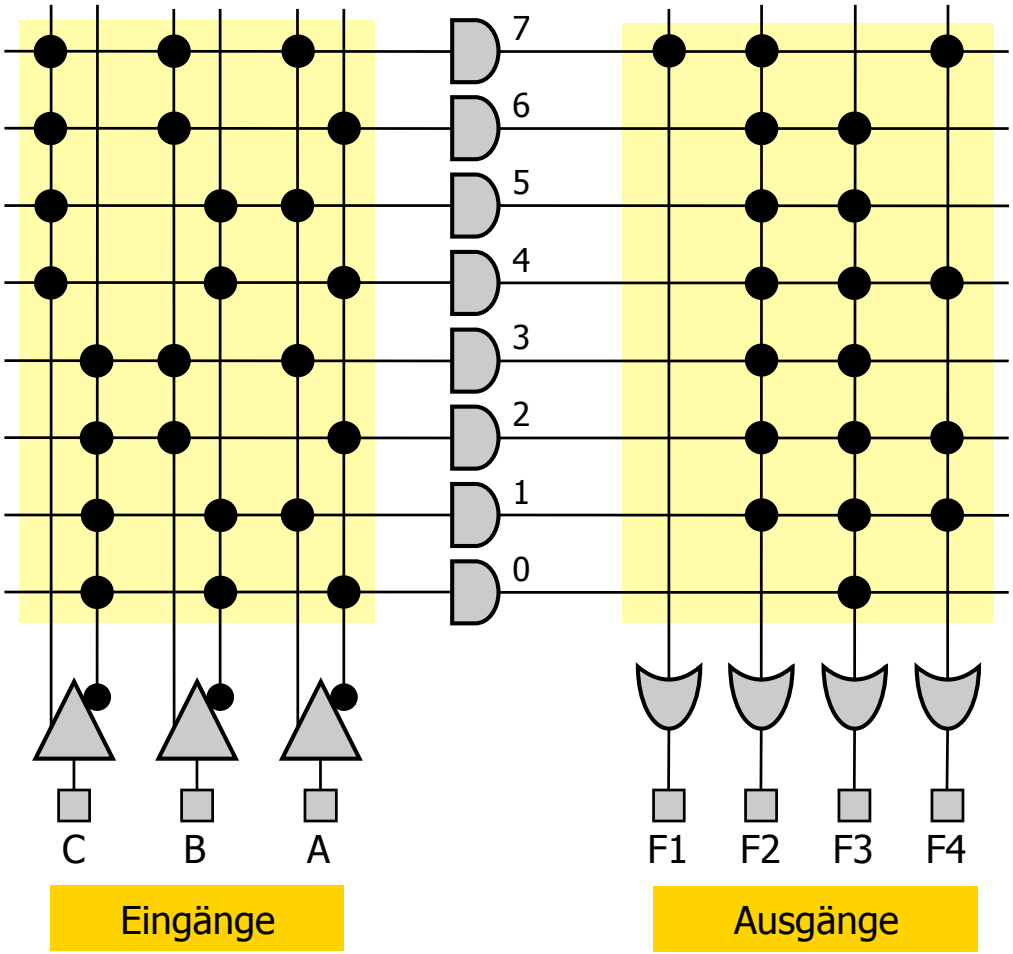


Beispiel für Codierung im PROM

- F1 = ABC
- F2 = A + B + C
- F3 = !(ABC) = !A + !B + !C
- F4 = A ⊕ B ⊕ C

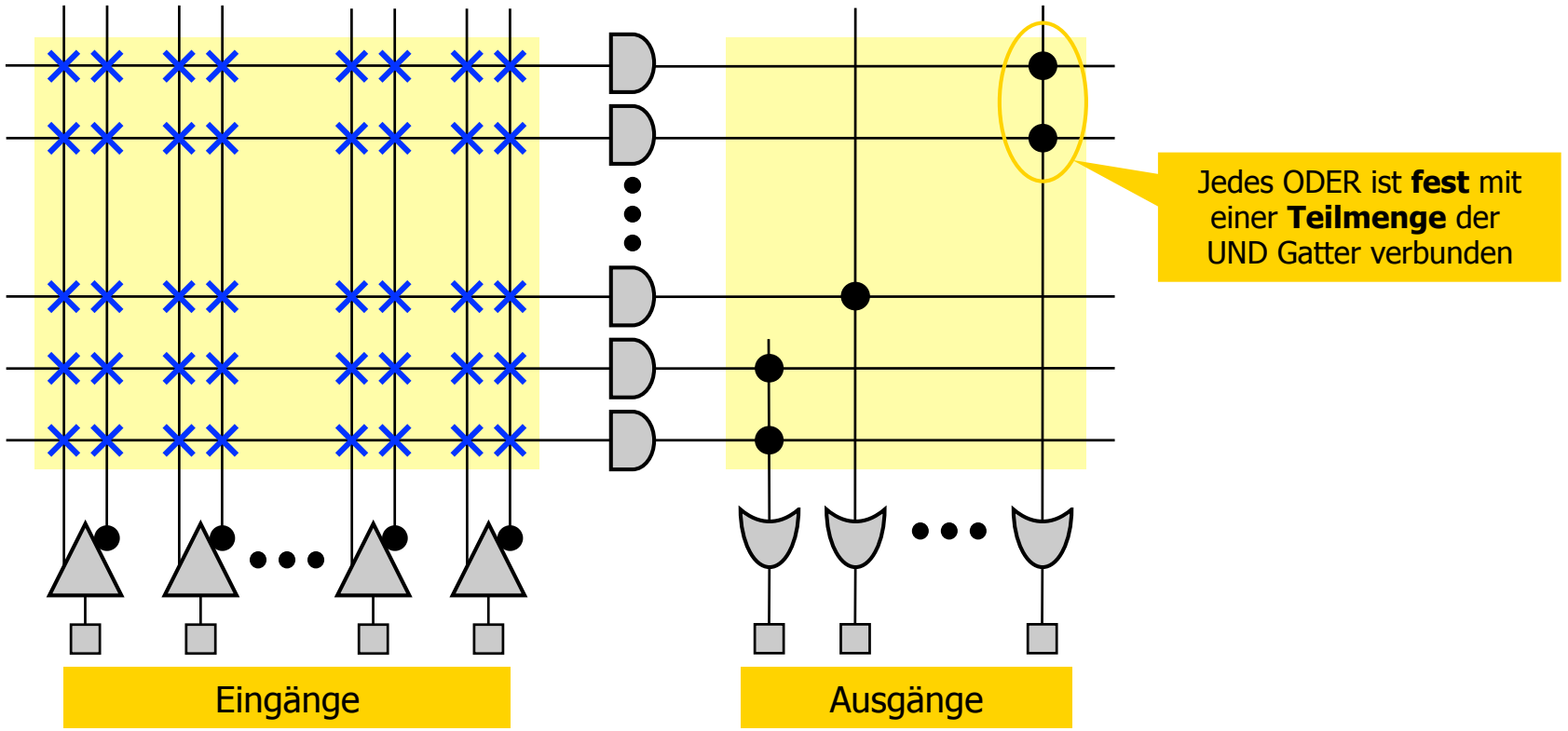
- Betrachte ein ‚3 x 4' ROM
  (3 Adressen, 4 Datenbits)

- Man sieht:
  Wegen der Verwendung von
  Mintermen ist das ODER
  schlecht (ineffizient) implemeniert.



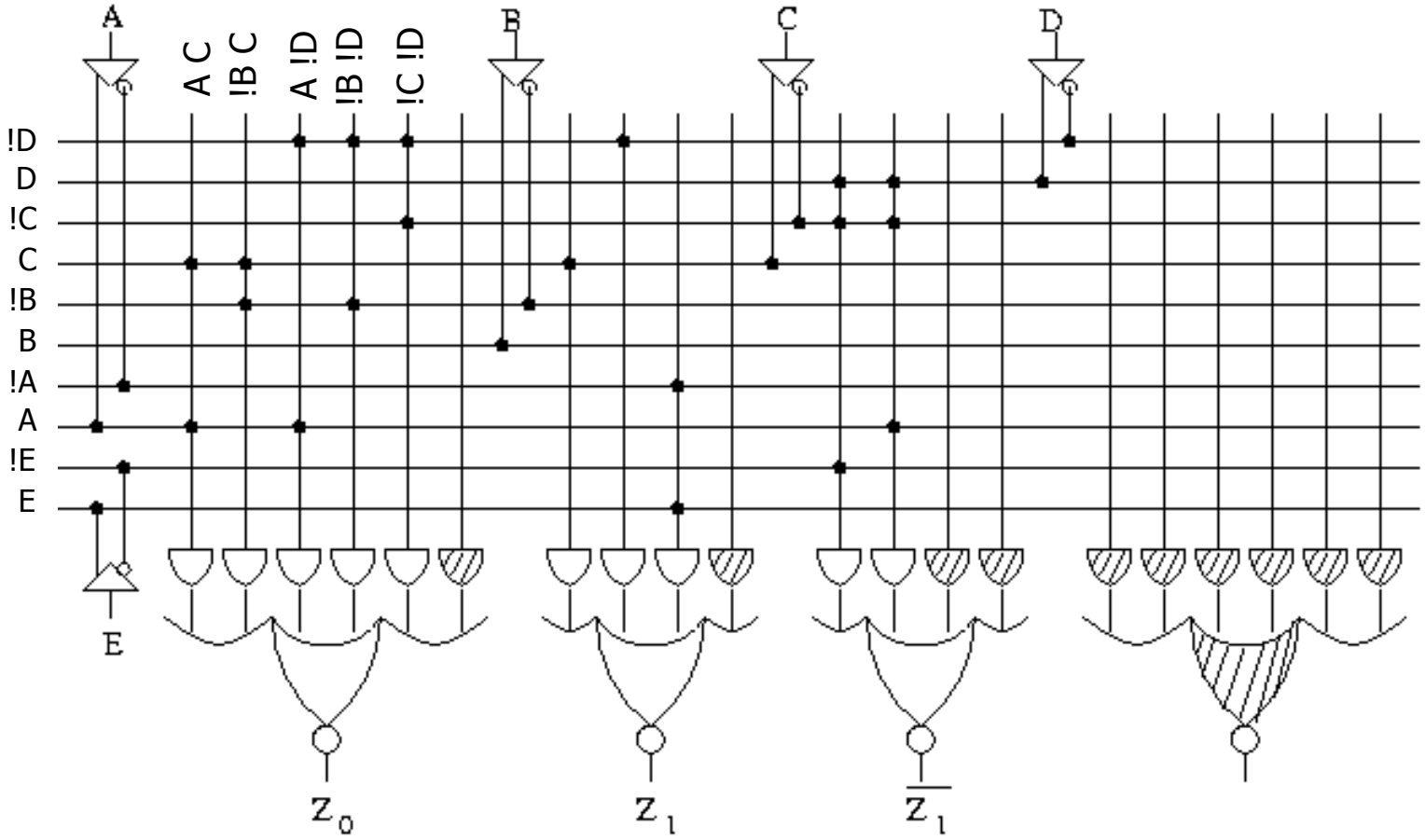
PAL

- Eingeführt von der Firma Monolithic Memories (MMI)
- Etwas eingeschränktere Struktur im Vergleich zum PLA:
  - Die ODER Struktur ist fest verdrahtet
  - Die Anzahl Eingänge zu den ODER-Gattern ist beschränkt (nicht immer für alle Ausgänge gleich)
  - Produktterme können nicht wiederverwendet werden
- PALs sind schneller als PLAs, da nur eine programmierbare Matrix durchlaufen werden muß (wenige ns)



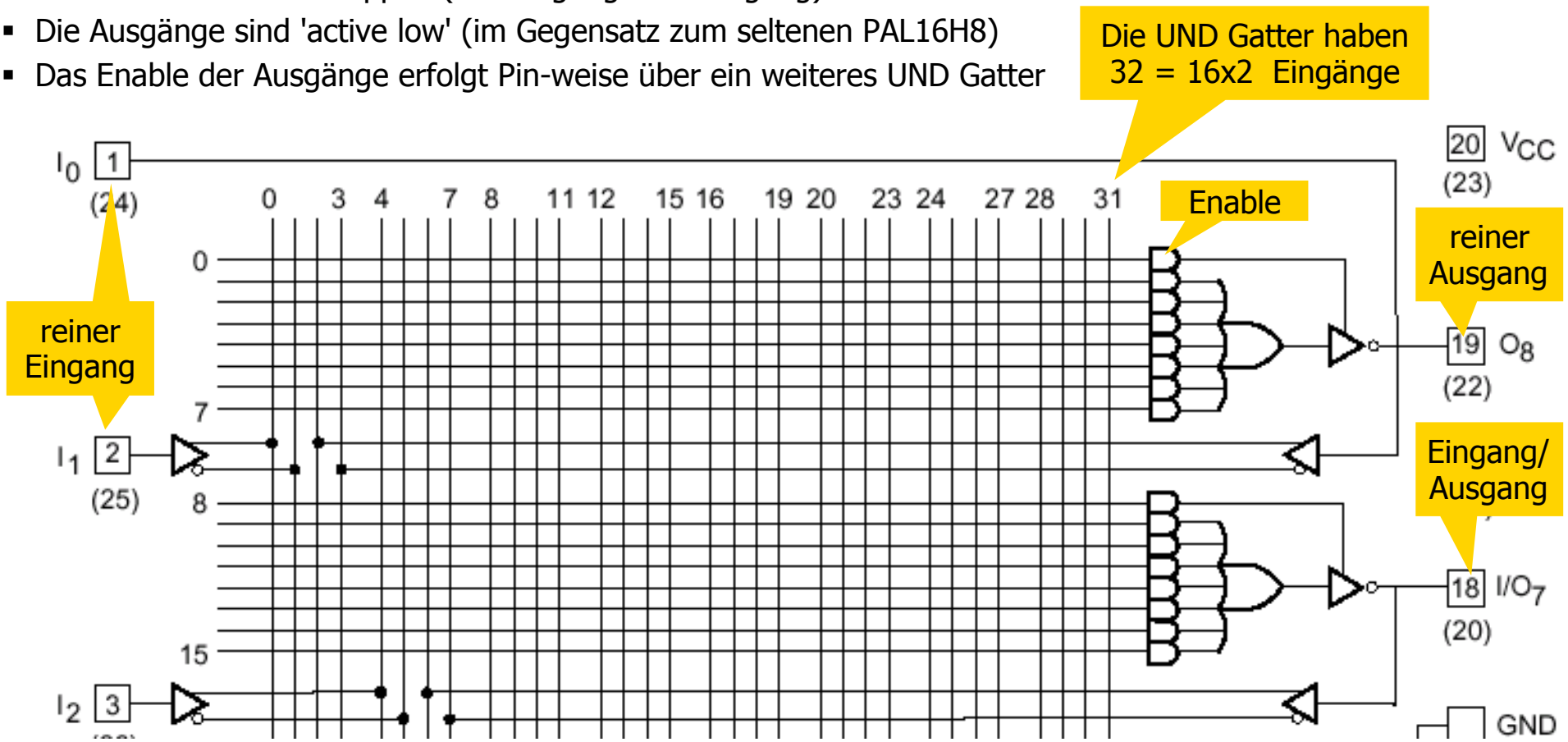
Beispiel: Funktion im PAL

- Implementiere
- Z0 = (A + !B + !C) (C + !D) = A C + !B C + A !D + !B !D + !C !D 'Summe von Produkten'
- Z1 = C + !D + !A · E
- !Z1 = !(C + !D + !A · E) = !C · D · !(!A E) = !C · D · (A + !E) = !C · D · A + !C · D · !E



Kombinatorische PALs

- Typischer Vertreter: PAL16L8
  - 16 Eingangsvariablen
  - 8 Ausgänge
- 7 Eingänge pro ODER Gatter
- Manche Pins werden doppelt (als Ausgang oder Eingang) benutzt
- Die Ausgänge sind 'active low' (im Gegensatz zum seltenen PAL16H8)
- Das Enable der Ausgänge erfolgt Pin-weise über ein weiteres UND Gatter



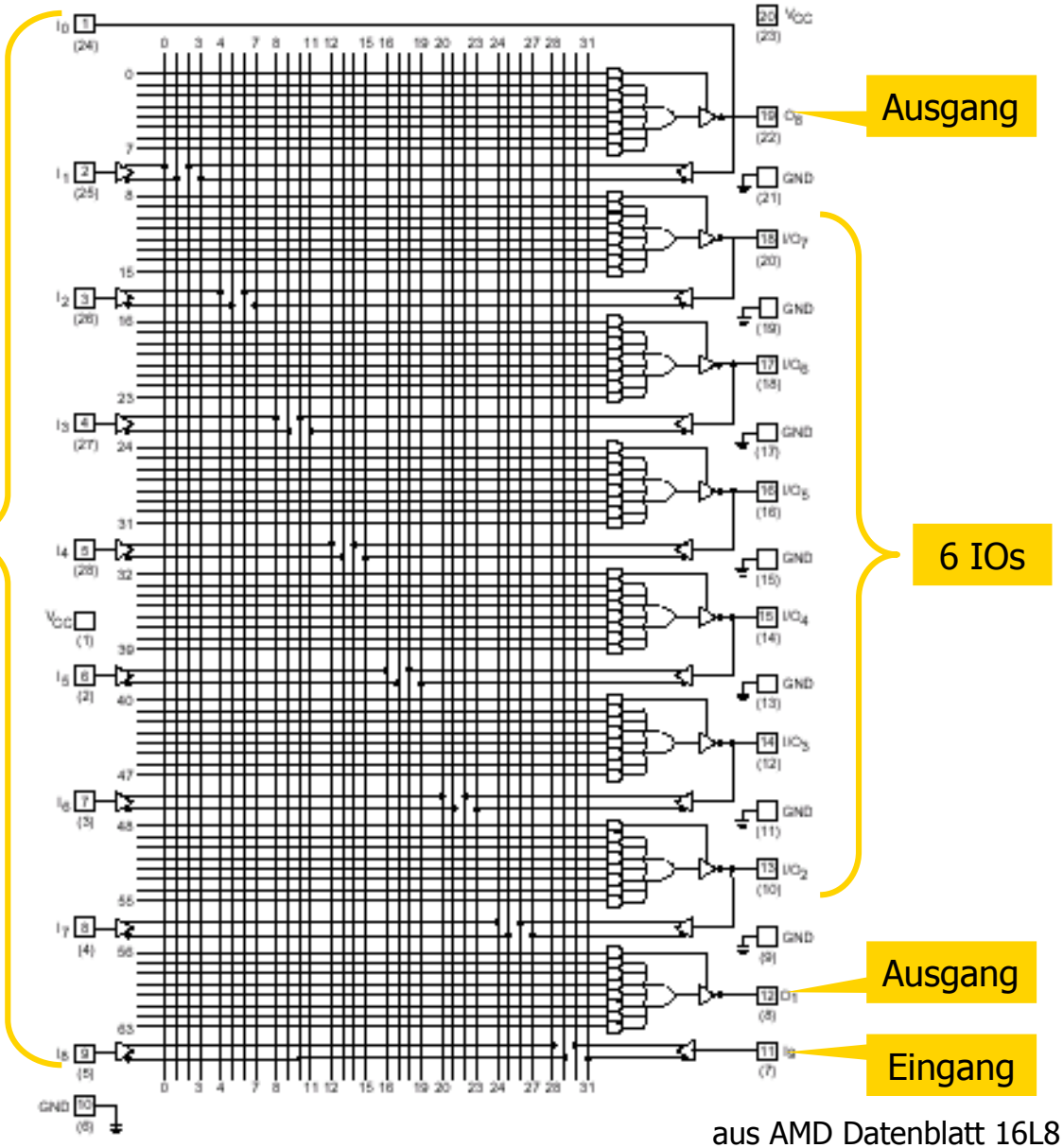
aus AMD Datenblatt 16L8

PAL 16L8

20 Pins:

- 10 reine Eingangspins
- 2 reine Ausgangspins
- 6 In/Out Pins
- 2 Versorgungen (Power, Ground)
- Gehäuse z.B. DIP20

- Also maximal Funktionen von 16 Eingangsvariablen
- Jedes ODER hat hier gleich viele Eingänge



aus AMD Datenblatt 16L8

PAL 16L8: Spezifikationen

- Beispiel für Datenblatt (AMD, 1996)
- ICC = 210mA !

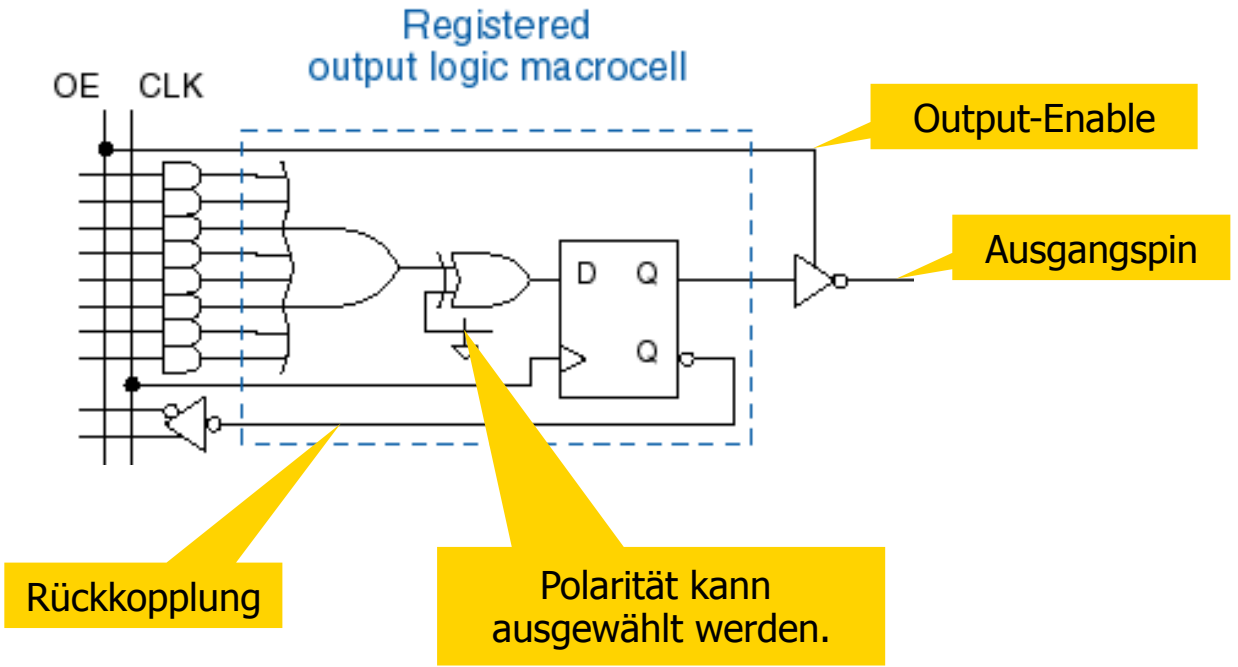
Commercial: 0^0C...+75^0C, Industrial: -40^0C..+85^0C

SWITCHING CHARACTERISTICS over COMMERCIAL operating ranges (Note 2)

Parameter Symbol	Parameter Description				-5 Min (Note 3)	-5 Max	-4 Min (Note 3)	-4 Max	Unit
t_{PD}	Input or Feedback to Combinatorial Output			16L8, 16R8, 16R4	1	5	1	4.5	ns
t_S	Setup Time from Input or Feedback to Clock			16R8, 16R6, 16R4	4.5		4.5		ns
t_H	Hold Time				0		0		ns
t_{CO}	Clock to Output				1	4.0	1	3.5	ns
t_{SKEWR}	Skew Between Registered Outputs (Note 4)					1		0.5	ns
t_{WL}	Clock Width	LOW			4		4		ns
t_{WH}		HIGH			4		4		ns
f_{MAX}	Maximum Frequency (Note 5)	External Feedback	$1/(t_S + t_{CO})$		117		125		MHz
		Internal Feedback (f_{CNT})	$1/(t_S + t_{CF})$ (Note 6)		125		125		MHz
		No Feedback	$1/(t_{WH} + t_{WL})$		125		125		MHz
t_{PZX}	$\overline{OE}$ to Output Enable				1	6.5	1	6.5	ns
t_{PXZ}	$\overline{OE}$ to Output Disable				1	5	1	5	ns
t_{EA}	Input to Output Enable Using Product Term Control			16L8, 16R6, 16R4	2	6.5	2	6.5	ns
t_{ER}	Input to Output Disable Using Product Term Control				2	5	2	5	ns

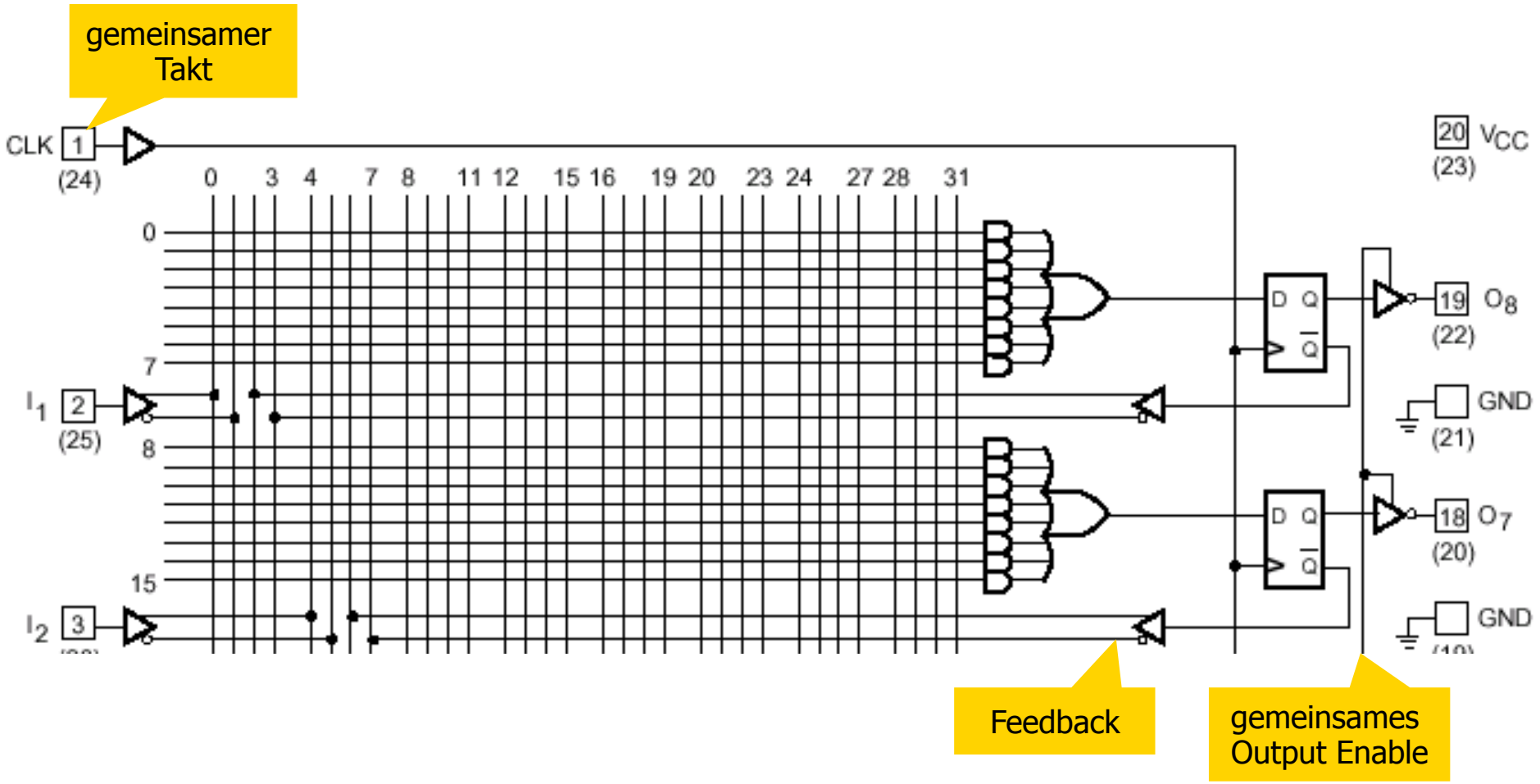
Ausgangszellen

- Verschiedene Varianten haben Flipflops am Ausgang, so dass auch Zustandsmaschinen implementiert werden können
- Die Ausgangswerte werden zurückgekoppelt
- Im gezeigten Beispiel können die Ausgänge nur über einen speziellen Chip-Pin aktiviert werden. In anderen Architekturen kann das Enable Signal durch eine Verknüpfung von Eingängen erzeugt werden.
- Manchmal kann die Polarität umgedreht werden. Damit vereinfachen sich manche logische Funktionen:
  A + B + C + D (4 UND Terme) = ! (!A · !B · !C · !D) (nur ein UND Term!)



PAL16R8 mit Registern

- 16 Eingänge zu den UND Gattern
- 8 Eingänge, 8 Ausgänge, 20 Pins (+ OEb, CLK, GND, VDD)
- OEb ist (hier) ein globaler Pin



Beispiel Gray Zähler

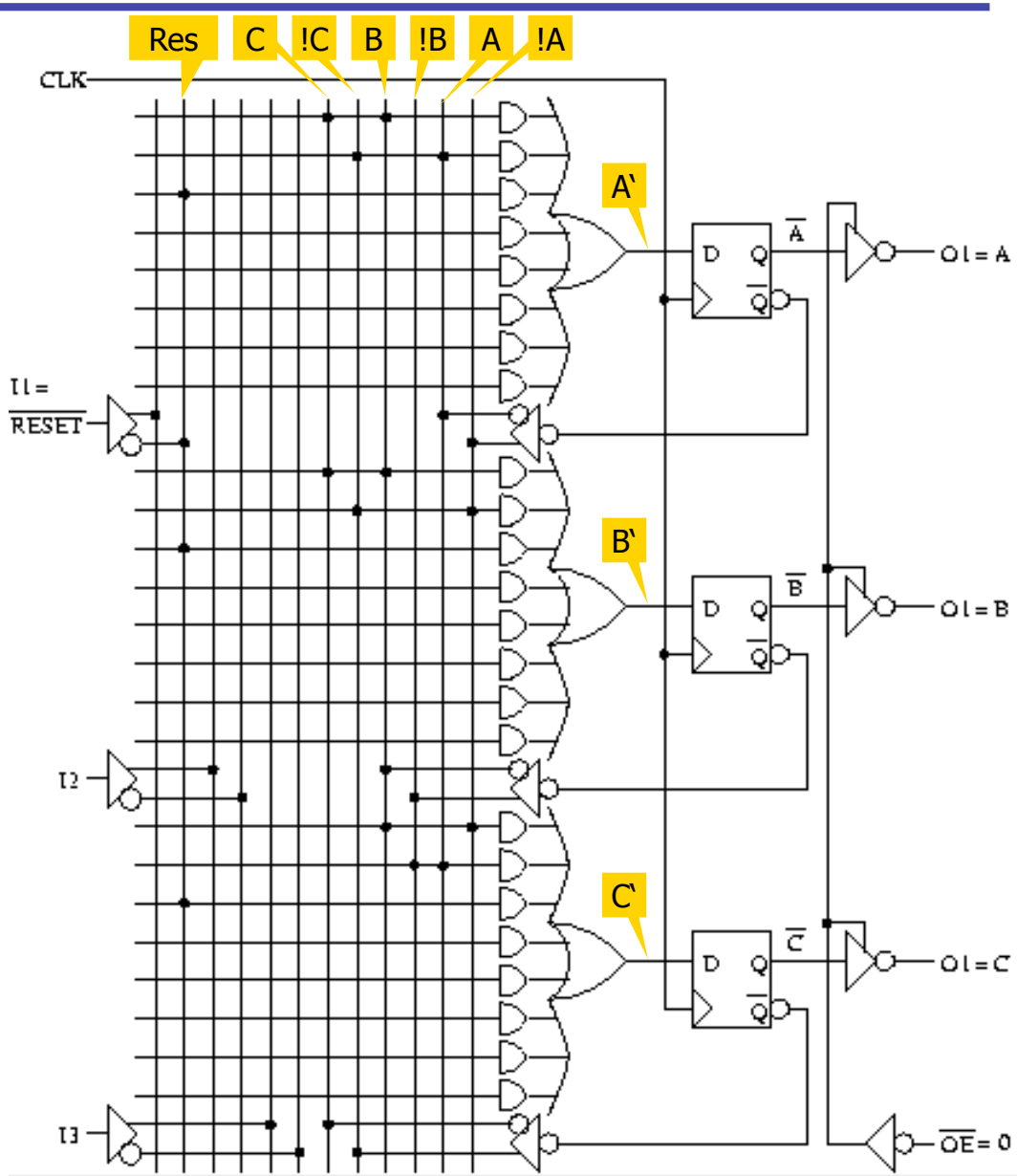
- PAL16R8 als 3 Bit Gray Zähler konfiguriert
- Nur ResetB Eingang ist benutzt

- A‘ = Res + C·B + A·!C
- B‘ = Res + C·B + !A·!C
- C‘ = Res + B·!A + A·!B

Zustand nach ResetB: 111

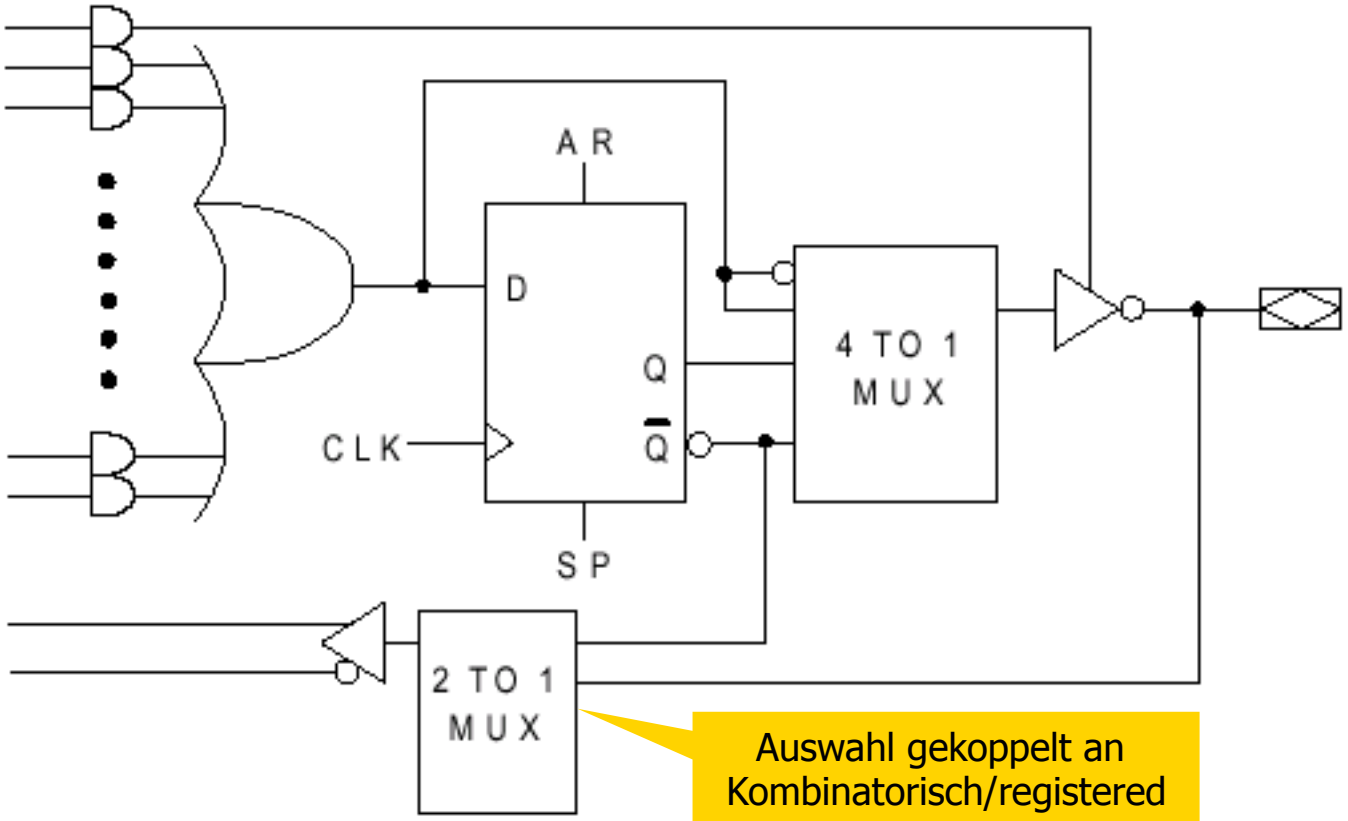
Dann:

A B C
1 1 1
1 1 0
1 0 0
1 0 1
0 0 1
0 0 0
0 1 0
0 1 1
1 1 1
...



Variable Ausgangsblöcke: GAL

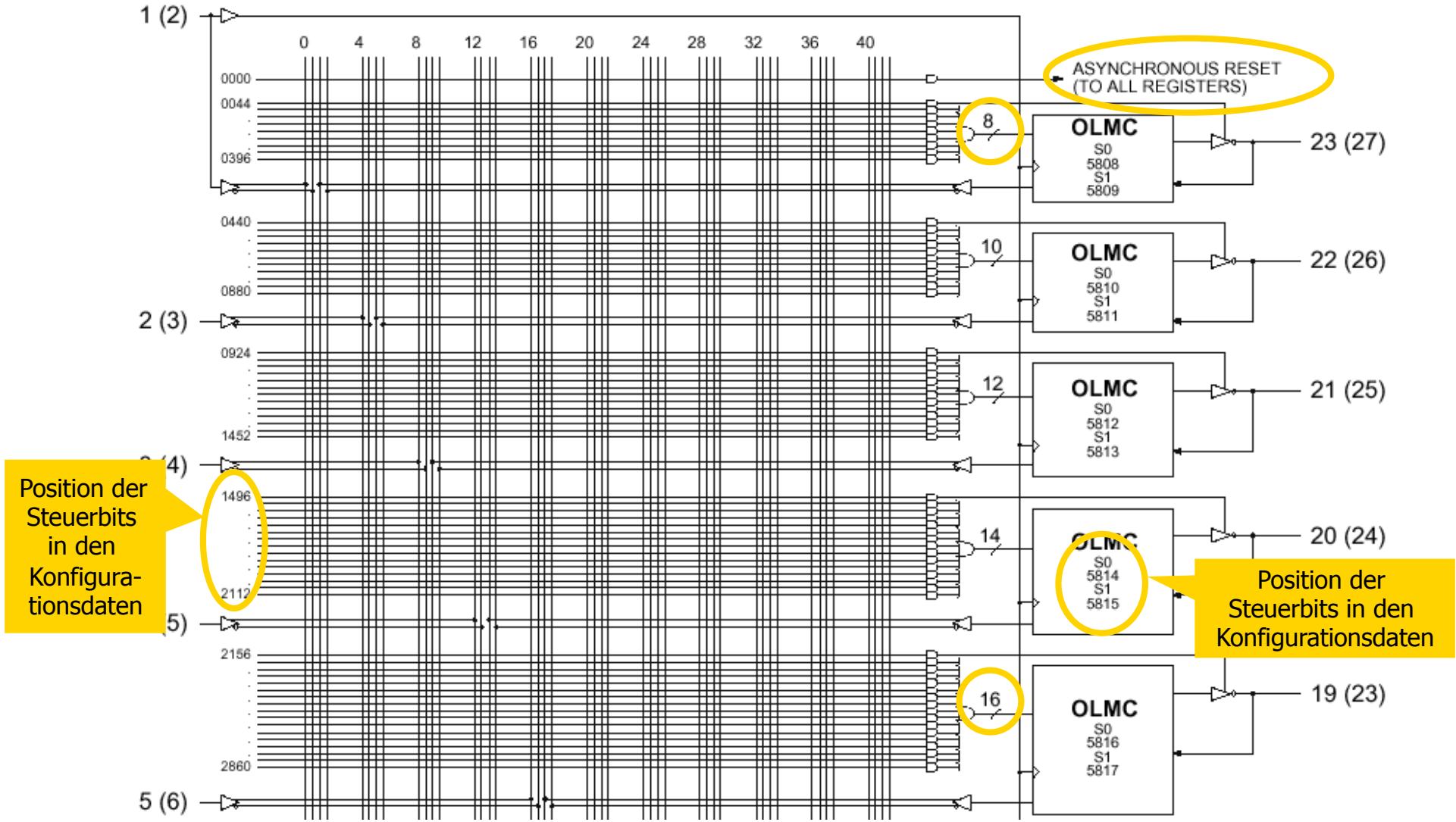
- **GAL**s (Firma Lattice) haben an jedem Pin eine **konfigurierbare Ausgangszelle** (hier 'OLMC')
- Sie kann als **kombinatorischer Ausgang** (active high/low), als **'registered' Ausgang** (active high/low) oder als **Eingang** genutzt werden.
- Das FF hat Setz- und Rücksetzeingänge, die (global) programmiert werden können
- Die Programmierung ist UV-löschbar, später auch elektrisch löschbar



GAL22V10 OUTPUT LOGIC MACROCELL (OLMC)

GAL22V10

- 22 Eingänge zu den UND Gattern, 10 Ausgänge mit Macrozellen, ODER mit 8-16 Eingängen
- Chip hat 24 Pins (GND, VDD, CLK/IN, 11xIn, 10xIO).



GAL22V10 – Ausschnitt Datenblatt

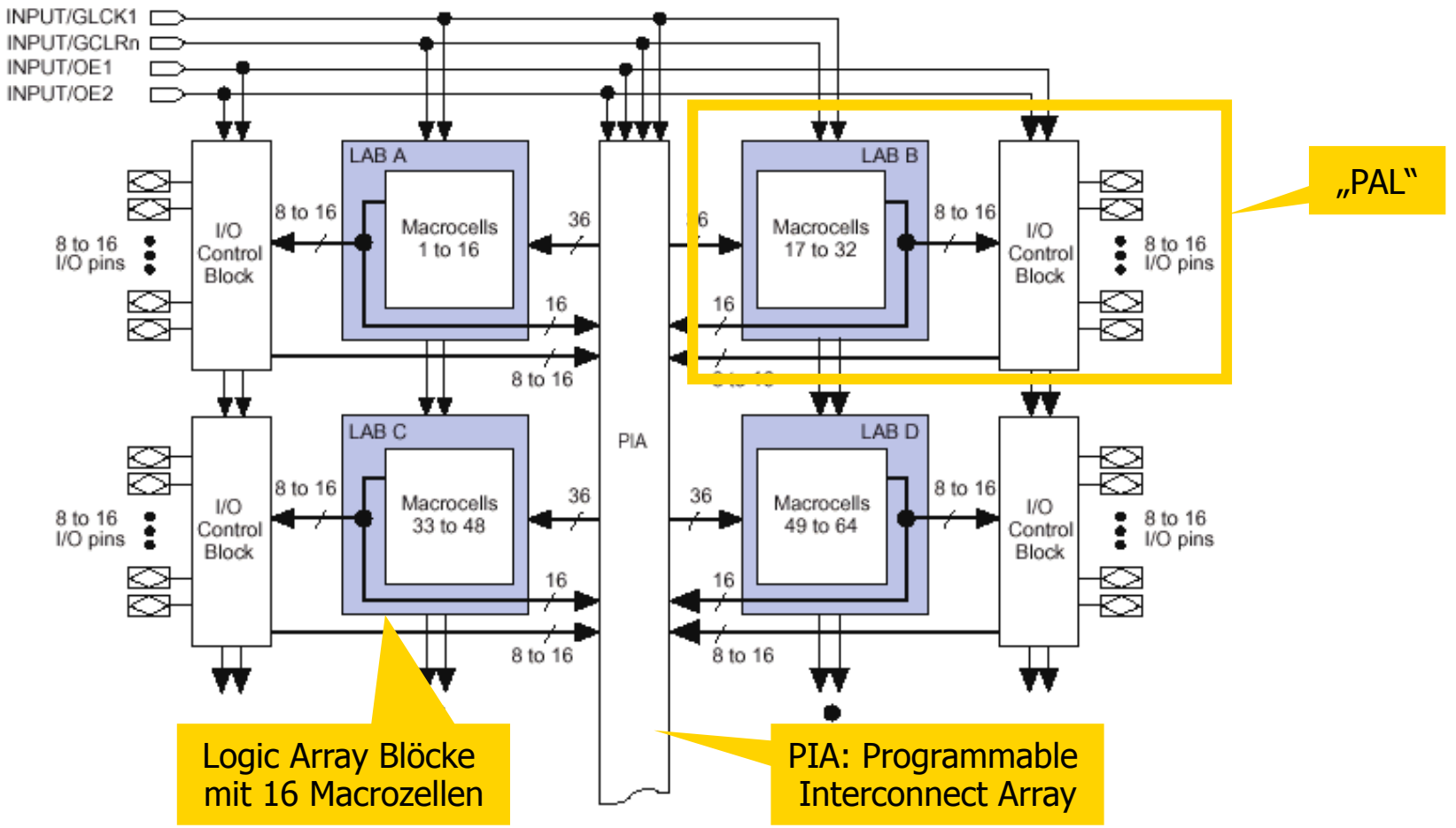
'speed grade'

PARAM	TEST COND.[1]	DESCRIPTION	COM -4 MIN.	COM -4 MAX.	COM -5 MIN.	COM -5 MAX.	COM/IND -7 MIN.	COM/IND -7 MAX.	UNITS
t_{pd}	A	Input or I/O to Combinatorial Output	1	4	1	5	1	7.5	ns
t_{co}	A	Clock to Output Delay	1	3.5	1	4	1	4.5	ns
t_{cf}[2]	—	Clock to Feedback Delay	—	2.5	—	3	—	3	ns
t_{su}	—	Setup Time, Input or Fdbk before Clk↑	2.5	—	3	—	4.5	—	ns
t_{h}	—	Hold Time, Input or Fdbk after Clk↑	0	—	0	—	0	—	ns
f_{max}[3]	A	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	167	—	142.8	—	111	—	MHz
	A	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	200	—	166	—	133	—	MHz
	A	Maximum Clock Frequency with No Feedback	250	—	200	—	166	—	MHz
t_{wh}	—	Clock Pulse Duration, High	2	—	2.5	—	3	—	ns
t_{wl}	—	Clock Pulse Duration, Low	2	—	2.5	—	3	—	ns
t_{en}	B	Input or I/O to Output Enabled	1	5	1	6	1	7.5	ns
t_{dis}	C	Input or I/O to Output Disabled	1	5	1	5.5	1	7.5	ns
t_{ar}	A	Input or I/O to Asynch. Reset of Reg.	1	4.5	1	5.5	1	9	ns
t_{arw}	—	Asynch. Reset Pulse Duration	4.5	—	4.5	—	7	—	ns
t_{arr}	—	Asynch. Reset to Clk↑ Recovery Time	3	—	4	—	5	—	ns
t_{spr}	—	Synch. Preset to Clk↑ Recovery Time	3	—	4	—	5	—	ns

CPLD

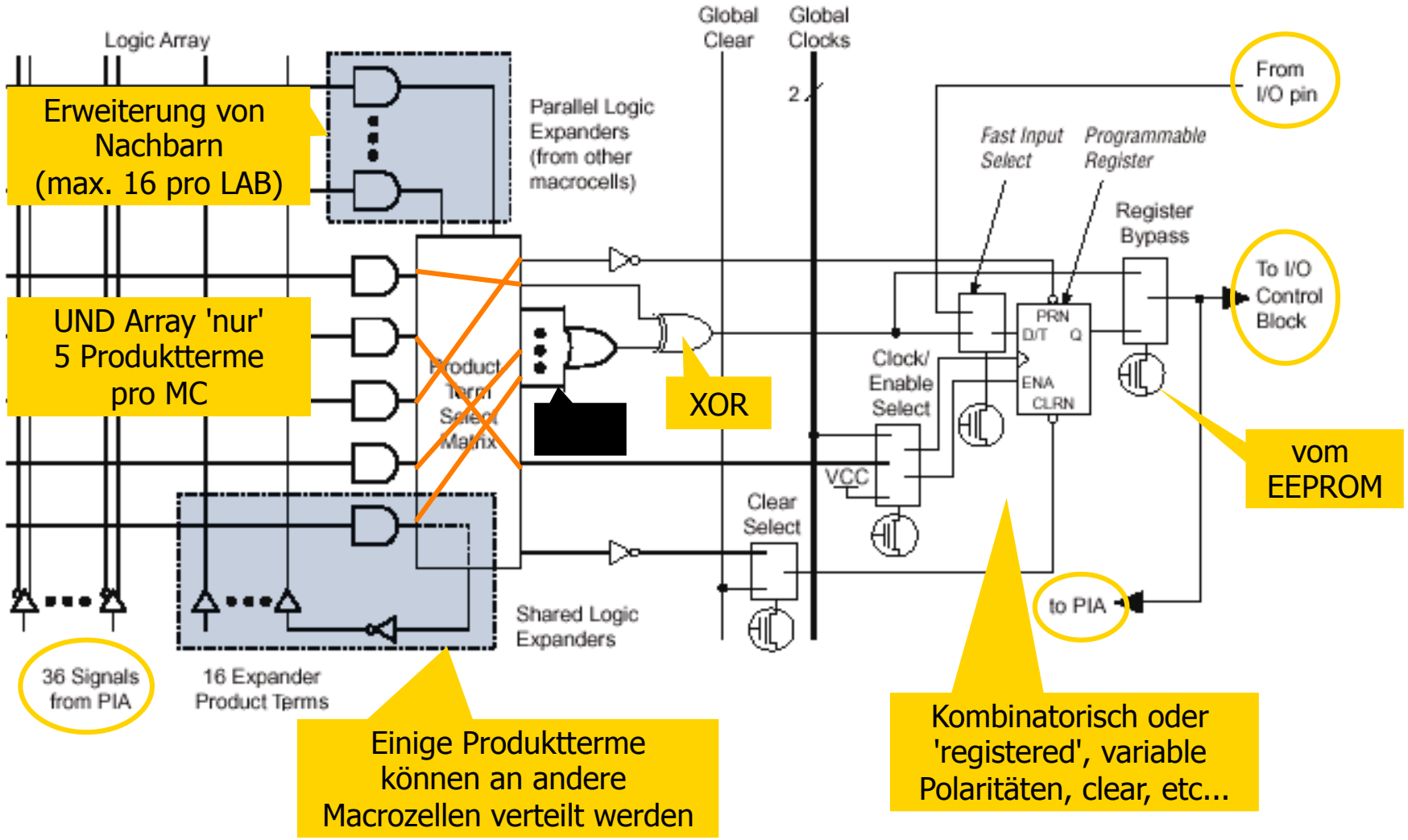
CPLD – Beispiel MAX7000

- CPLD = 'Complex Programmable Logic Devices'
- Erster Ansatz: Viele PAL-ähnliche Blöcke (hier 'Logic Array Blocks' = 'LABs') auf einem Chip.
- Beispiel: Altera, MAX7000 – Familie
- Hier: Speicherung der Konfiguration in CMOS EEPROM (bis zu 100x programmierbar)



MAX7000 Macrozelle

- Sie sind in jedem 'Logic Array Block' (LAB) mehrfach (16x) vorhanden
- UND Array hat 5 Ausgänge. Die 'Product Term Select Matrix' verteilt sie auf ODER, XOR, set, reset, clk, etc.



MAX7000 Familie

Table 1. MAX 7000 Device Features

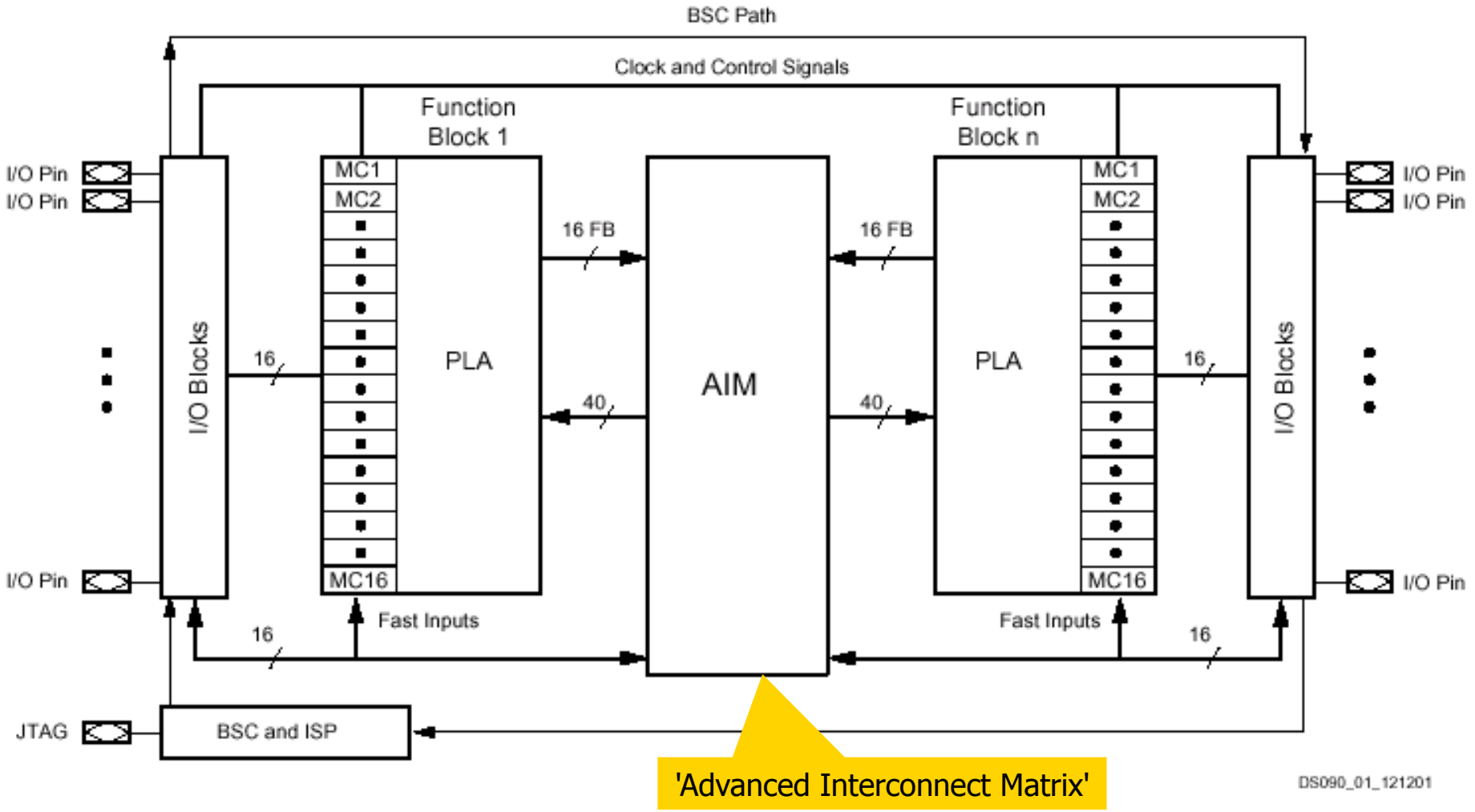
Feature	EPM7032	EPM7064	EPM7096	EPM7128E	EPM7160E	EPM7192E	EPM7256E
Usable gates	600	1,250	1,800	2,500	3,200	3,750	5,000
Macrocells	32	64	96	128	160	192	256
Logic array blocks	2	4	6	8	10	12	16
Maximum user I/O pins	36	68	76	100	104	124	164
t_{PD} (ns)	6	6	7.5	7.5	10	12	12
t_{SU} (ns)	5	5	6	6	7	7	7
t_{FSU} (ns)	2.5	2.5	3	3	3	3	3
t_{CO1} (ns)	4	4	4.5	4.5	5	6	6
f_{CNT} (MHz)	151.5	151.5	125.0	125.0	100.0	90.9	90.9

Table 5. MAX 7000 Maximum User I/O Pins *Note (1)*

Device	44-Pin PLCC	44-Pin PQFP	44-Pin TQFP	68-Pin PLCC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP	160-Pin PQFP	160-Pin PGA	192-Pin PGA	208-Pin PQFP	208-Pin RQFP
EPM7032	36	36	36									
EPM7032S	36		36									
EPM7064	36		36	52	68	68						
EPM7064S	36		36		68		68					
EPM7096				52	64	76						
EPM7128E					68	84		100				
EPM7128S					68	84	84 *(2)*	100				

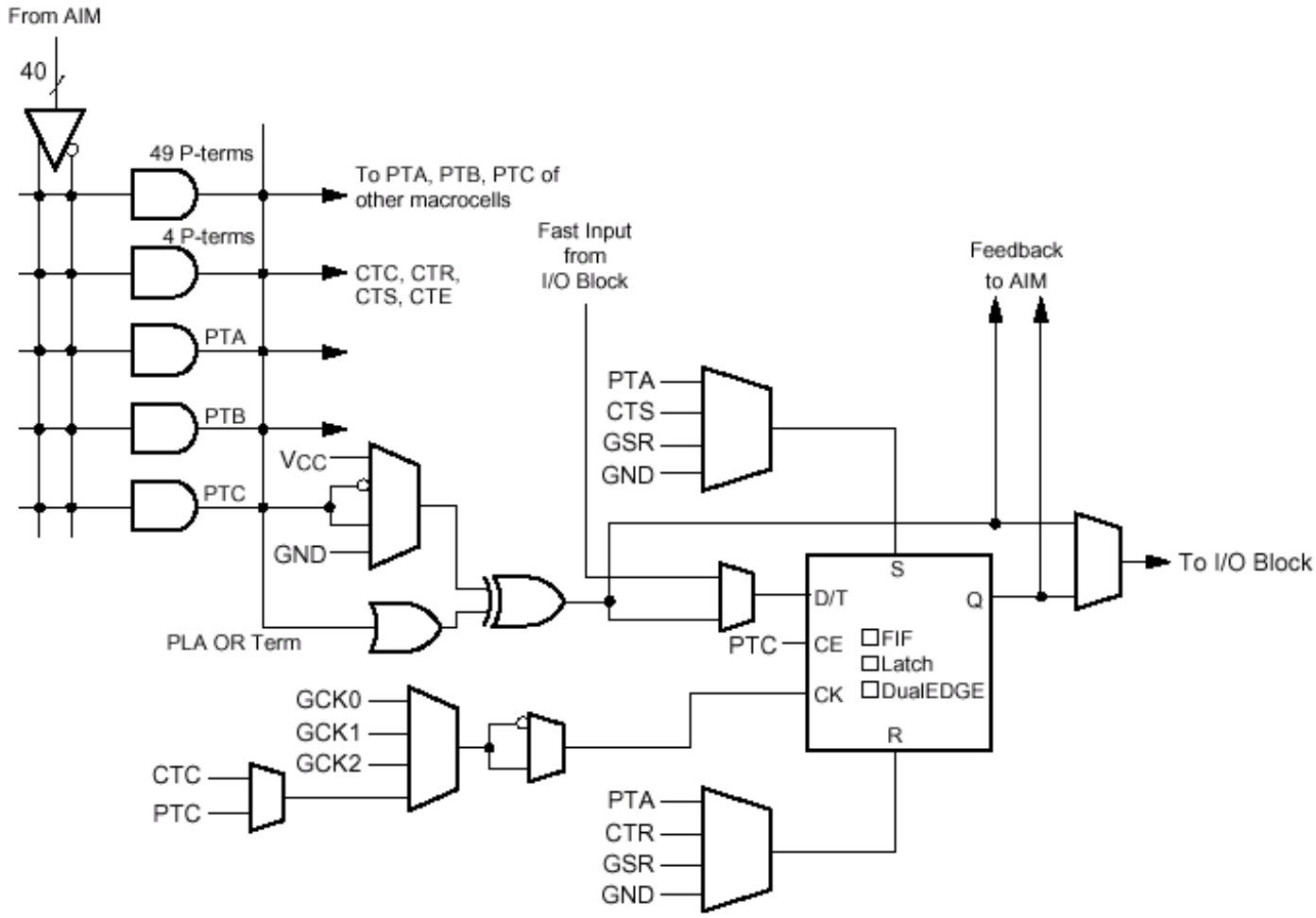
Im Vergleich: Xilinx CPLD Familie 'Coolrunner'

- Low Power CPLD (ehemals Philips). 'Traditional CPLD Architecture'
- 1.8V, 0.18µm Technologie, IO bis 3.3V, I_{stat} < 100µA (daher ‚Coolrunner')
- EEPROM, JTAG,..



Coolrunner Macrocell

- Sehr ähnlich wie MAX7000...



Weitere Blöcke/'Features'

JTAG (IEEE 1149.1)

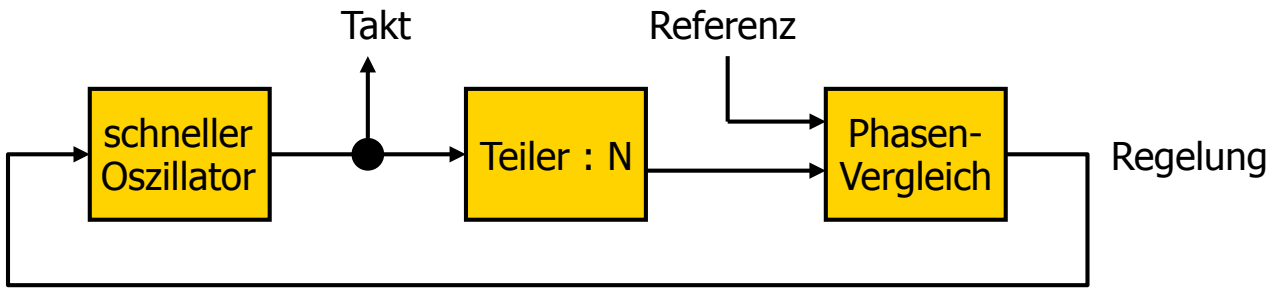
- Zum Programmieren und Testen haben sehr viele ICs heute ein **JTAG** interface '**J**oint **T**est **A**ction **G**roup':
- Alle Pins eines ICs werden (neben ihrer normalen Funktion) mit einem Schieberegister verbunden, so daß man beliebige **IO Muster schreiben und lesen** kann, ohne die Pins anzuschließen!
- Sehr populär zum Testen von ICs 'in Circuit' und zum Testen von Verbindungen auf PCBs

Embedded RAM

- Viele CPLDs bieten RAM-Blöcke unterschiedlicher Größe an, die ins Design eingebunden werden können
- Die Architektur ist meist programmierbar (8k x 1, 4k x 2, ...)
- Oft auch dual ported etc.

PLL

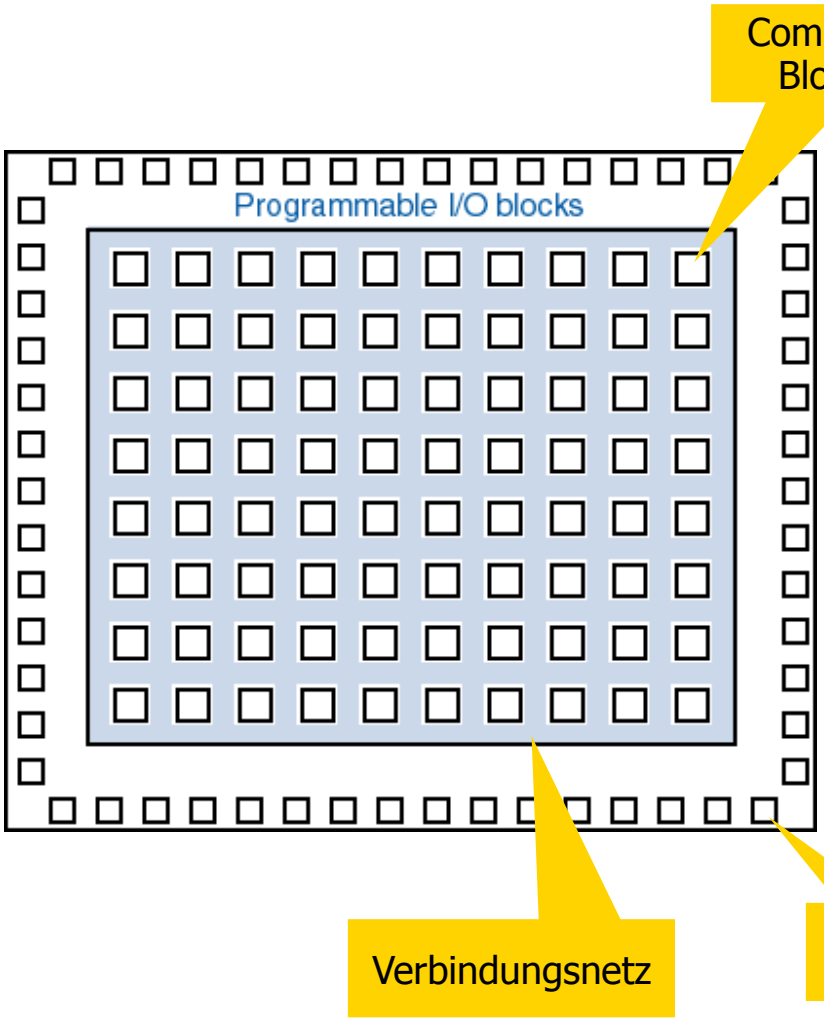
- Viele Designs können intern mit (mehreren) schnelleren Takten arbeiten. Zur Erzeugung aus einem 'langsamen', externen Takt werden Phase-Locked-Loops (PLLs) benutzt:
- Ein Oszillator erzeugt den 'schnellen' Takt. Dieser wird heruntergeteilt und mit dem 'langsamen', externen Referenztakt verglichen. Der 'schnelle' Oszillator wird nachgeregelt.

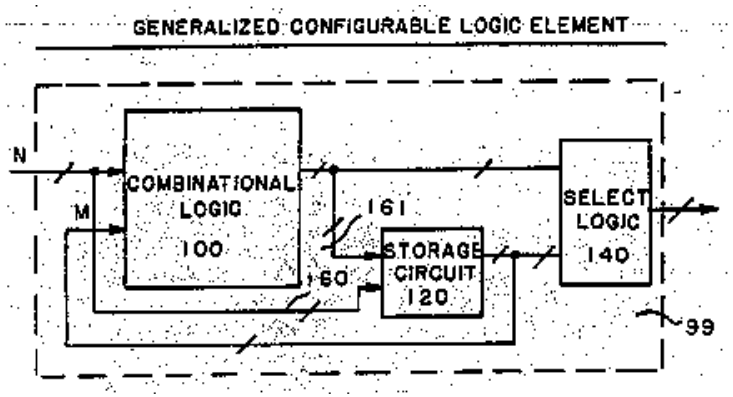


FPGA / LCA

Komplexere programmierbare Bausteine

- Ein etwas anderer Ansatz benutzt viele 'kleine' Logikblöcke (z.B. Xilinx, 1984)
- Sie werden durch ein Netz von programmierbaren Kreuzungen miteinander verbunden





Xilinx Patent 4,706,216
'Configurable Logic Element' (1984)

Beispiel Xilinx 4000-er Familie

- Die CLBs haben nur 'wenige' Eingänge (hier G1-G4 und F1-F4)
- Einige spezielle Funktionen, insbesondere für Carry-Funktionen

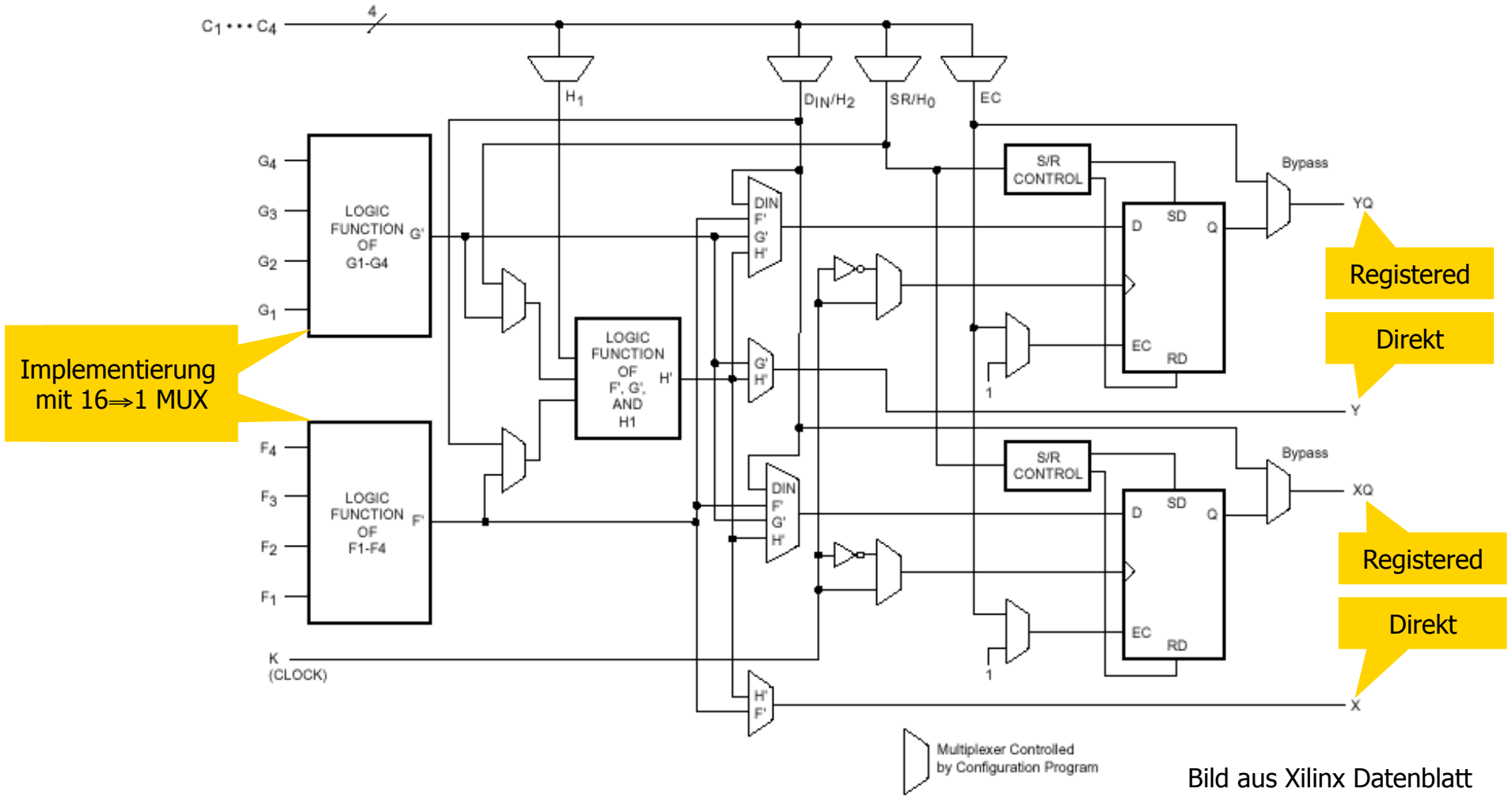
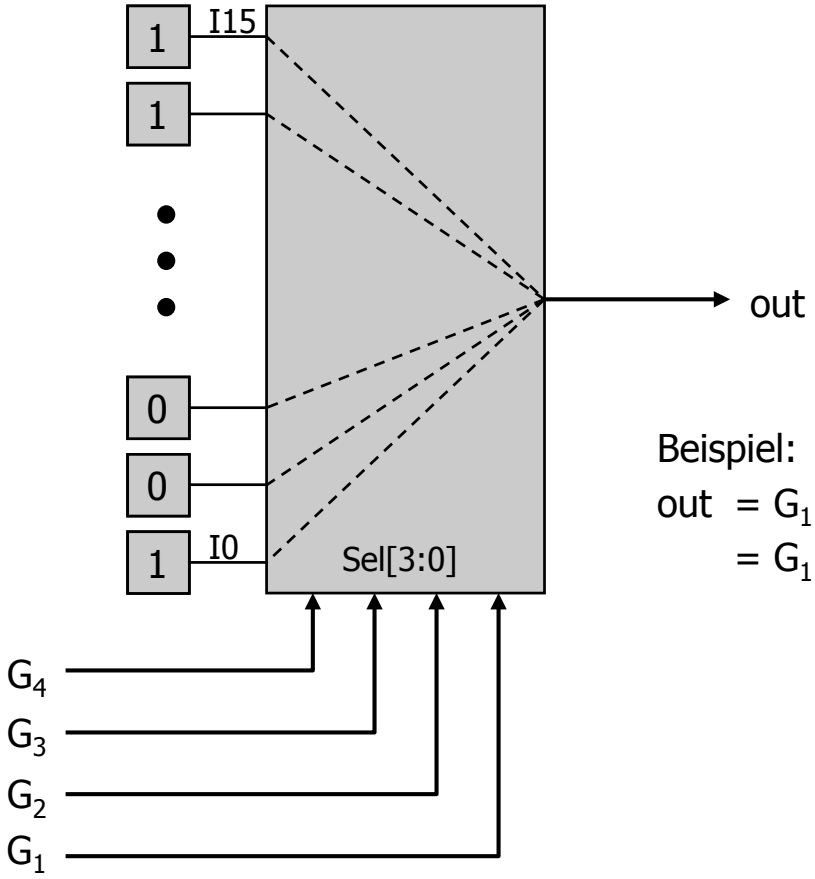


Bild aus Xilinx Datenblatt

Implementierung der ‚Logic Function' Blöcke

- Es soll eine beliebige Funktion von 4 Variablen, z.B. $G_4 \ldots G_1$ programmierbar sein.
- Es gibt dafür $2^{2^4} = 2^{16} = 65536$ Möglichkeiten
- Lösung : 16⇒1 MUX. Das Muster an den 16 Eingängen ist programmierbar, die Eingangsvariablen selektieren einen der 16 Eingänge



Beispiel:

$$\text{out} = G_1\,G_2\,G_3\,G_4 + !G_1\,G_2\,G_3\,G_4 + !G_1\,!G_2\,!G_3\,!G_4$$
$$= G_1\,G_2\,G_3 + !G_1\,!G_2\,!G_3\,!G_4$$

Verfügbare Logikfunktion

- **Zwei beliebige** Funktionen von **4** Eingangsvariablen $F_4...F_1$ und $G_4...G_1$
- **Eine beliebige** Funktion von **5** Variablen E_9,$E_4...E_1$ ($E_4...E_1$für E_9=0 in F codiert, für E_9=1 in G. E_9 wählt aus)
- **Manche** Funktionen von bis zu 9 Variablen $E_9...E_1$ (s. Bild)

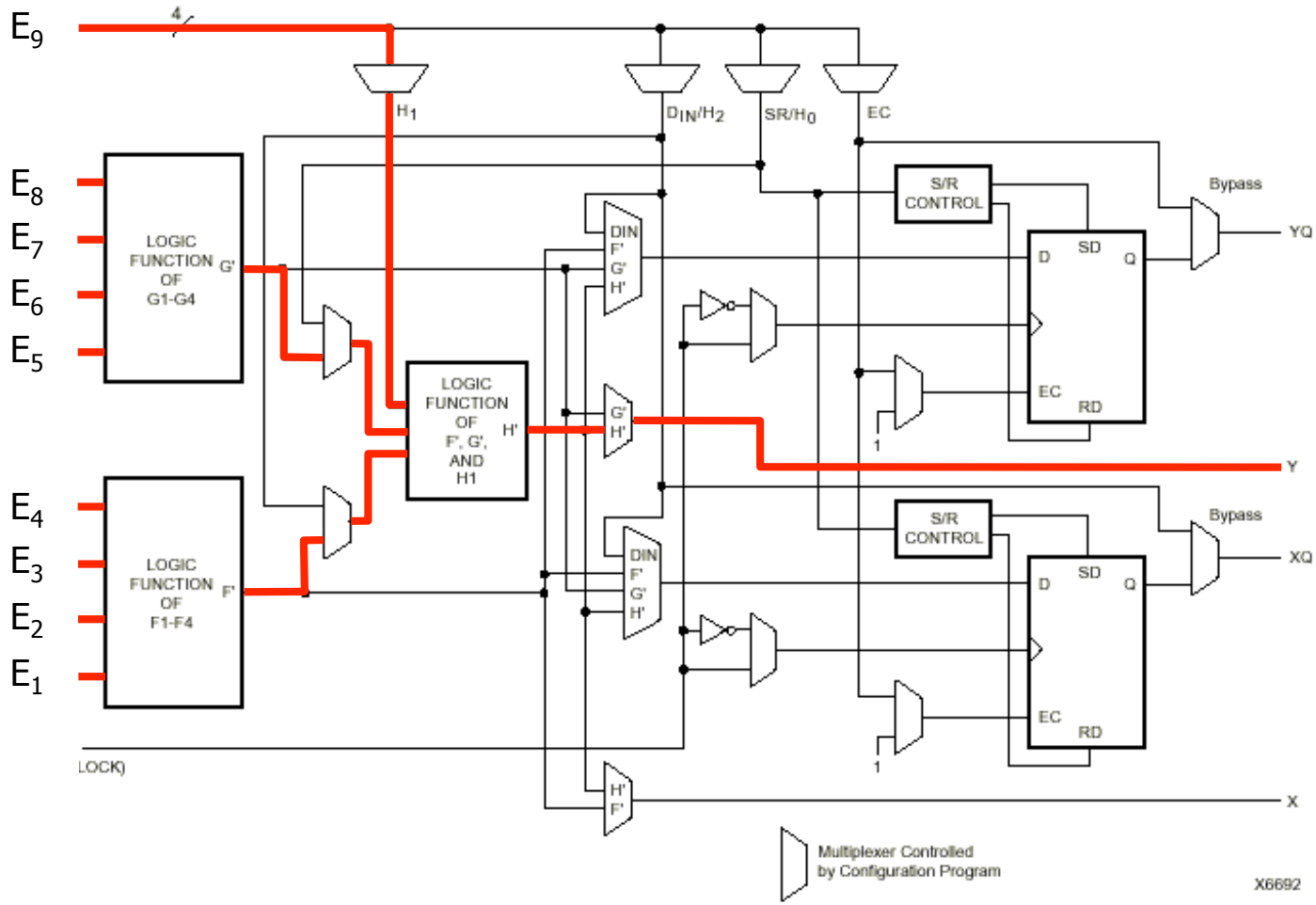


Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

Gemischte Logikfunktion im CLB

- Hier sind zwei getaktete Funktionen von je 4 Variablen (rot, grün) und eine kombinatorische von 2 Variablen implementiert (blau).

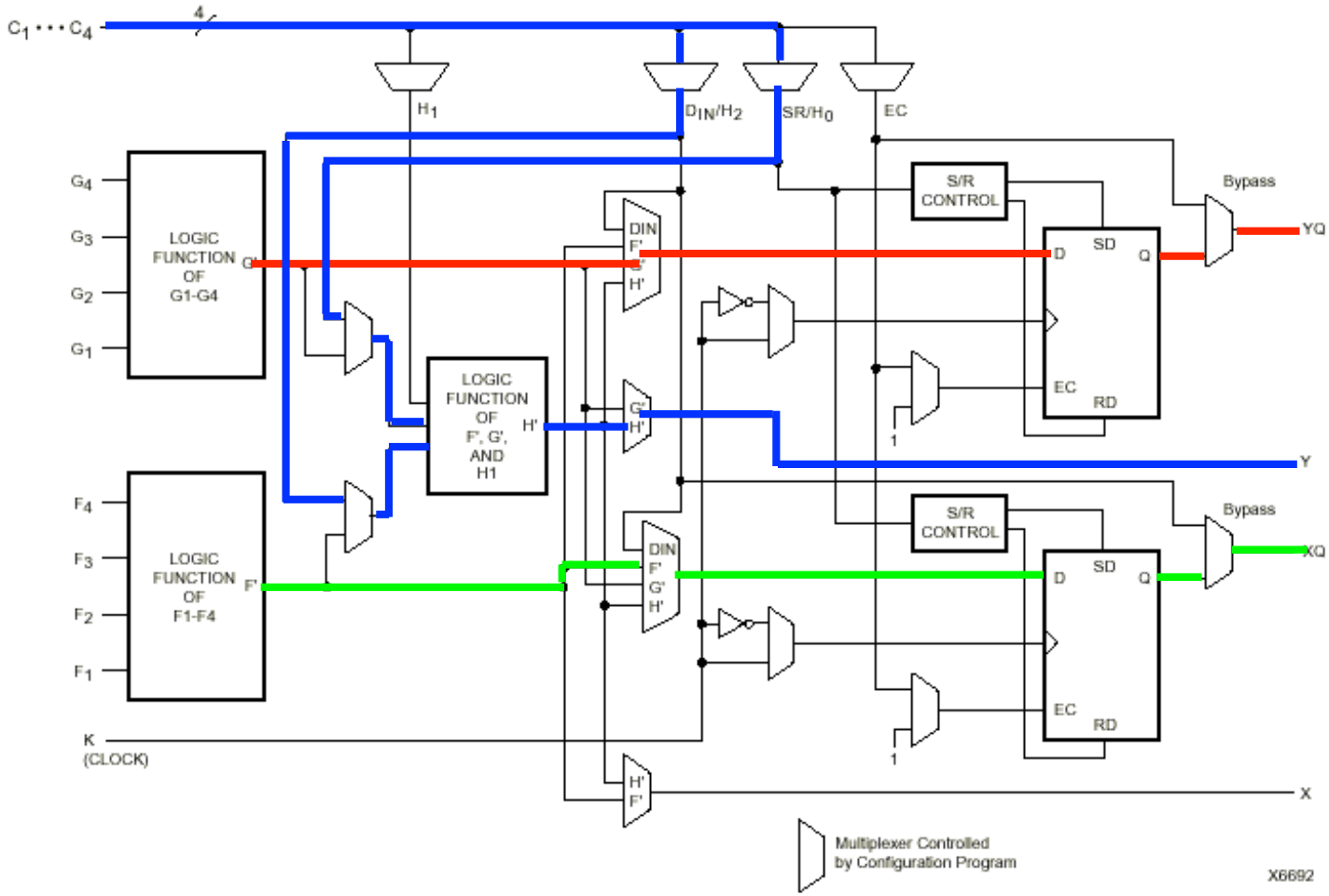
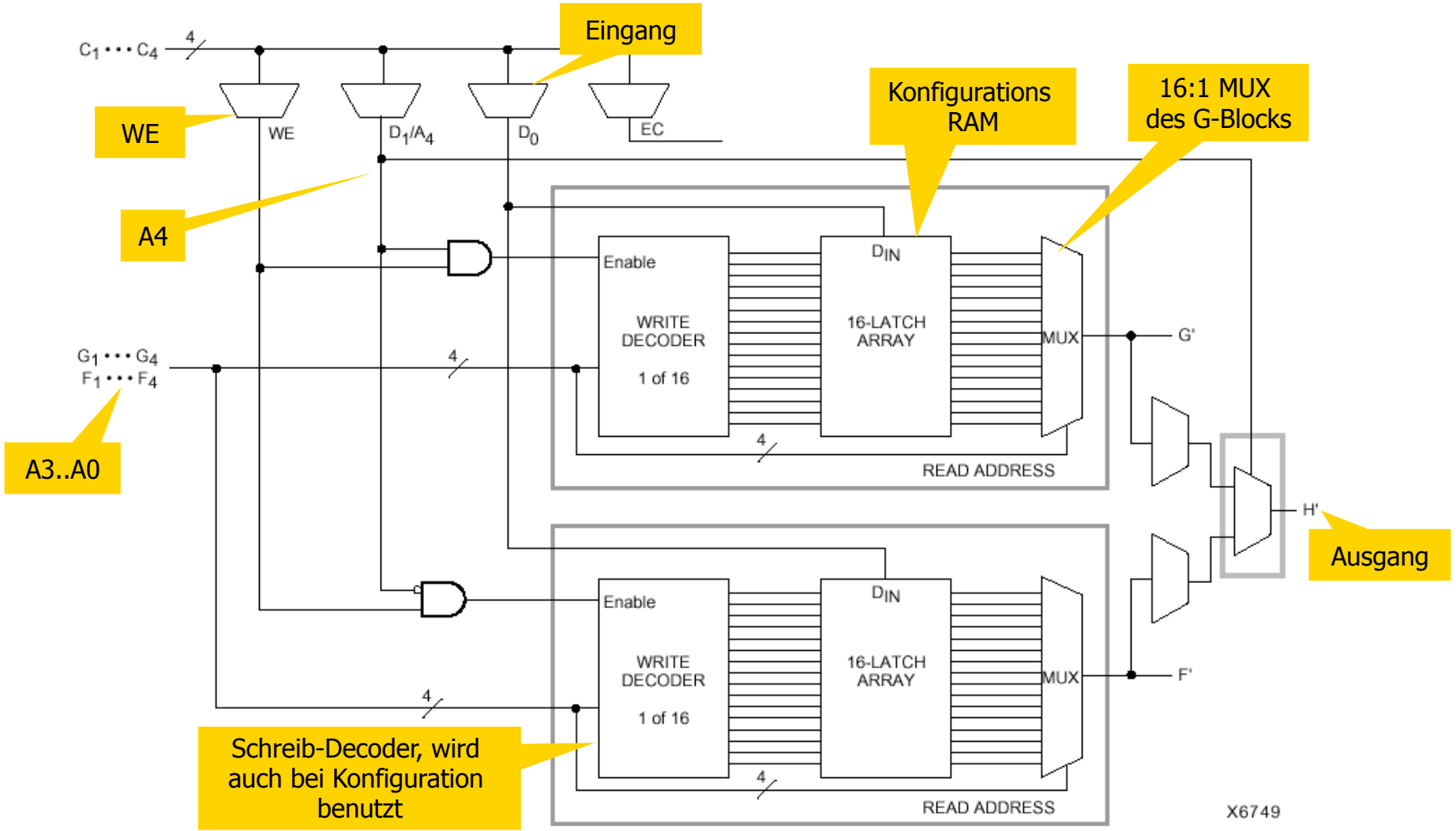


Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

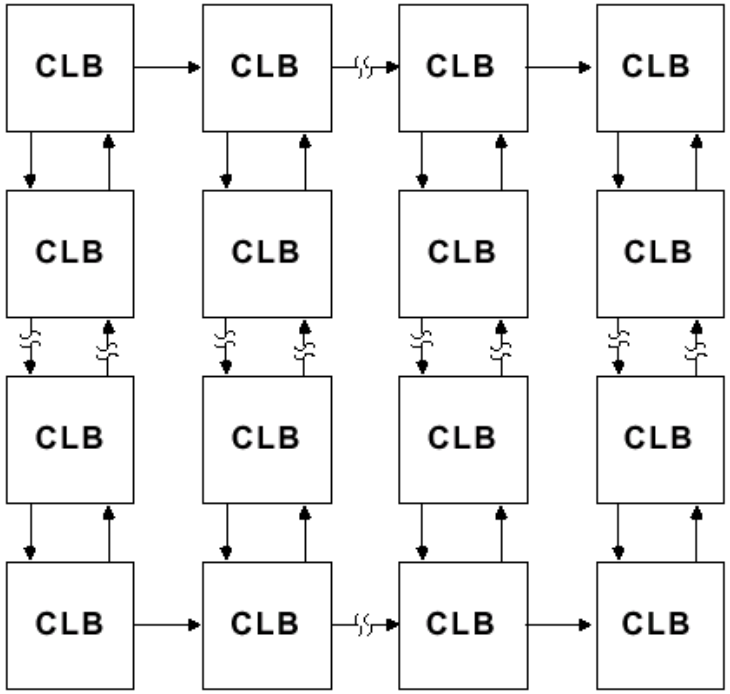
XC4000 CLB (und Folgende) als RAM

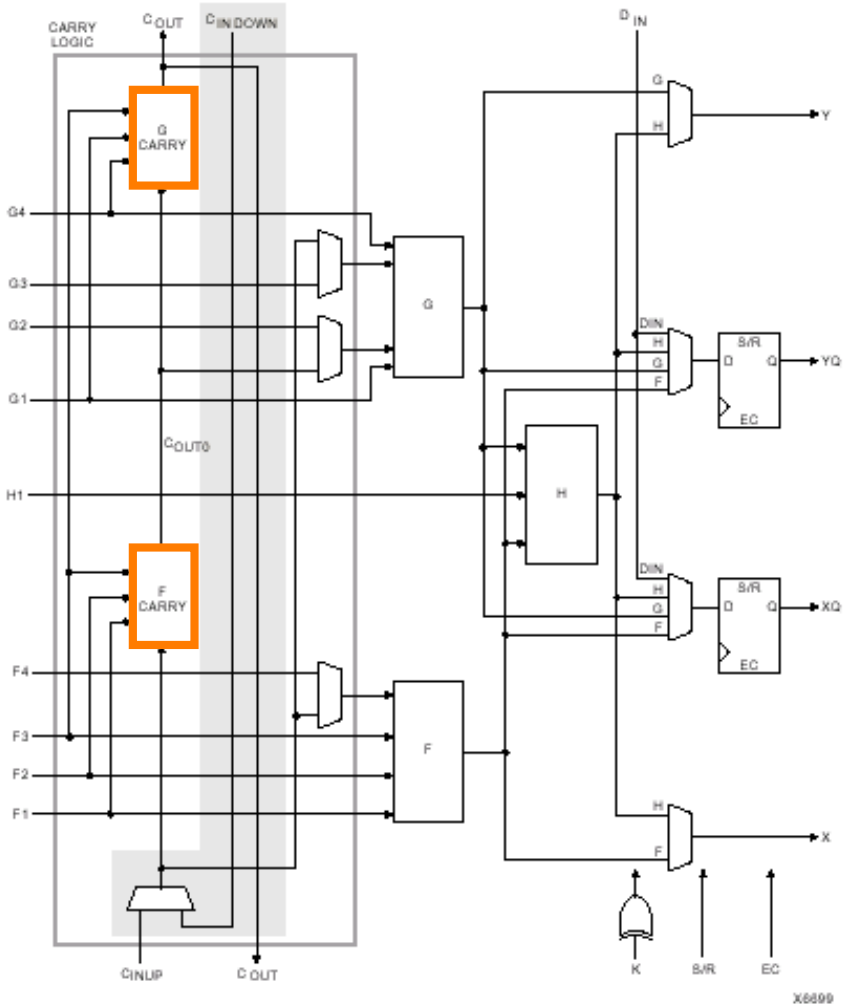
- Die XC4000 – Familie hat kein dediziertes RAM.
- Die MUXe der CLBs können aber als 16x2 oder 32x1 RAM (s. Bild) konfiguriert werden



XC4000 (und Folgende) Fast Carry Logic

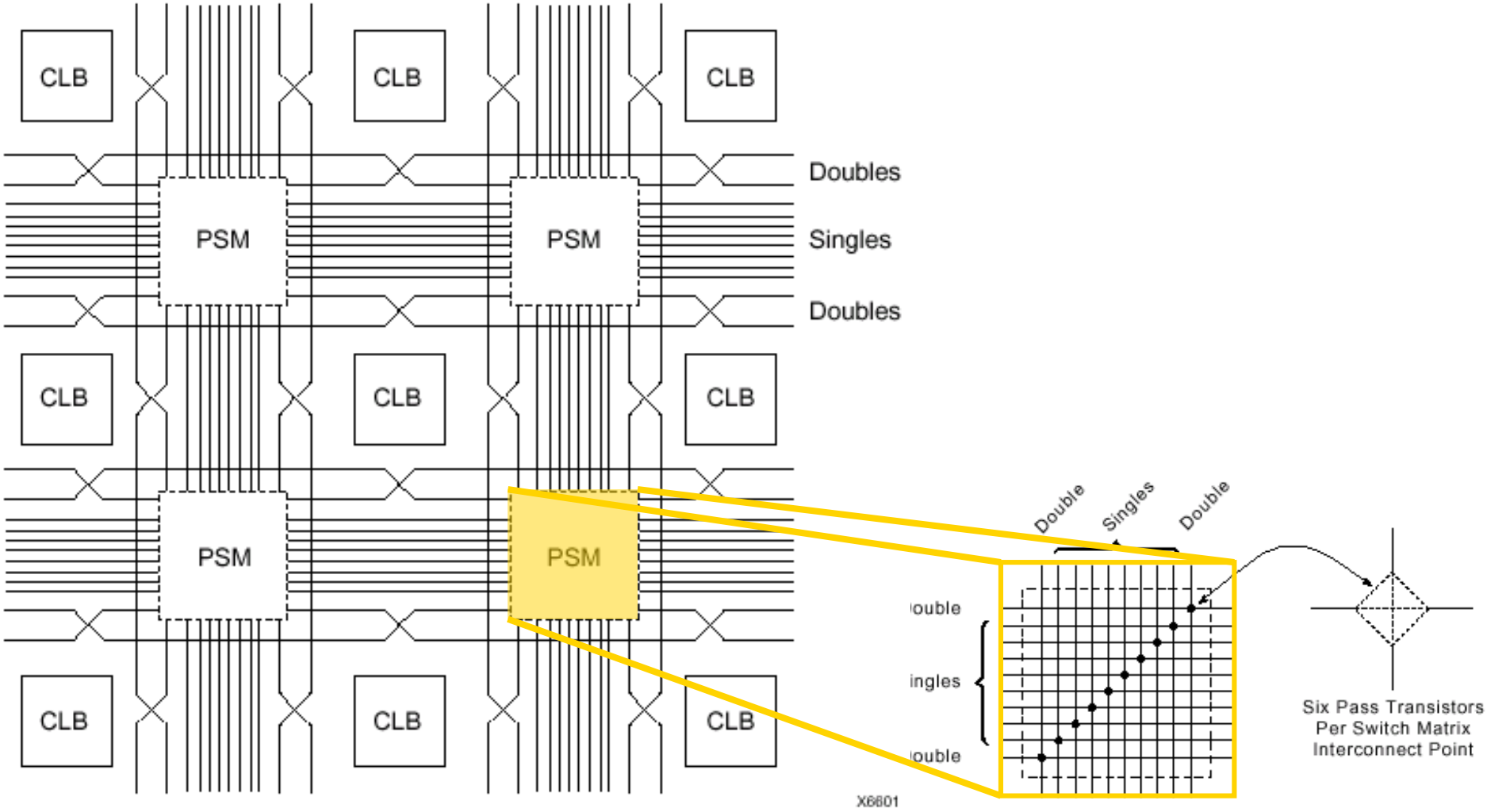
- Eine spezielle, flexible 'Carry'-Logik zum Aufbau von Zählern und Addierern ist vorhanden.
- Schnelle Carry-Signale verbinden benachbarte CLBs
- Laut Datenblatt können damit Zähler bis 32 Bit Breite ohne schaltungstechnische Optimierungen implementiert werden.



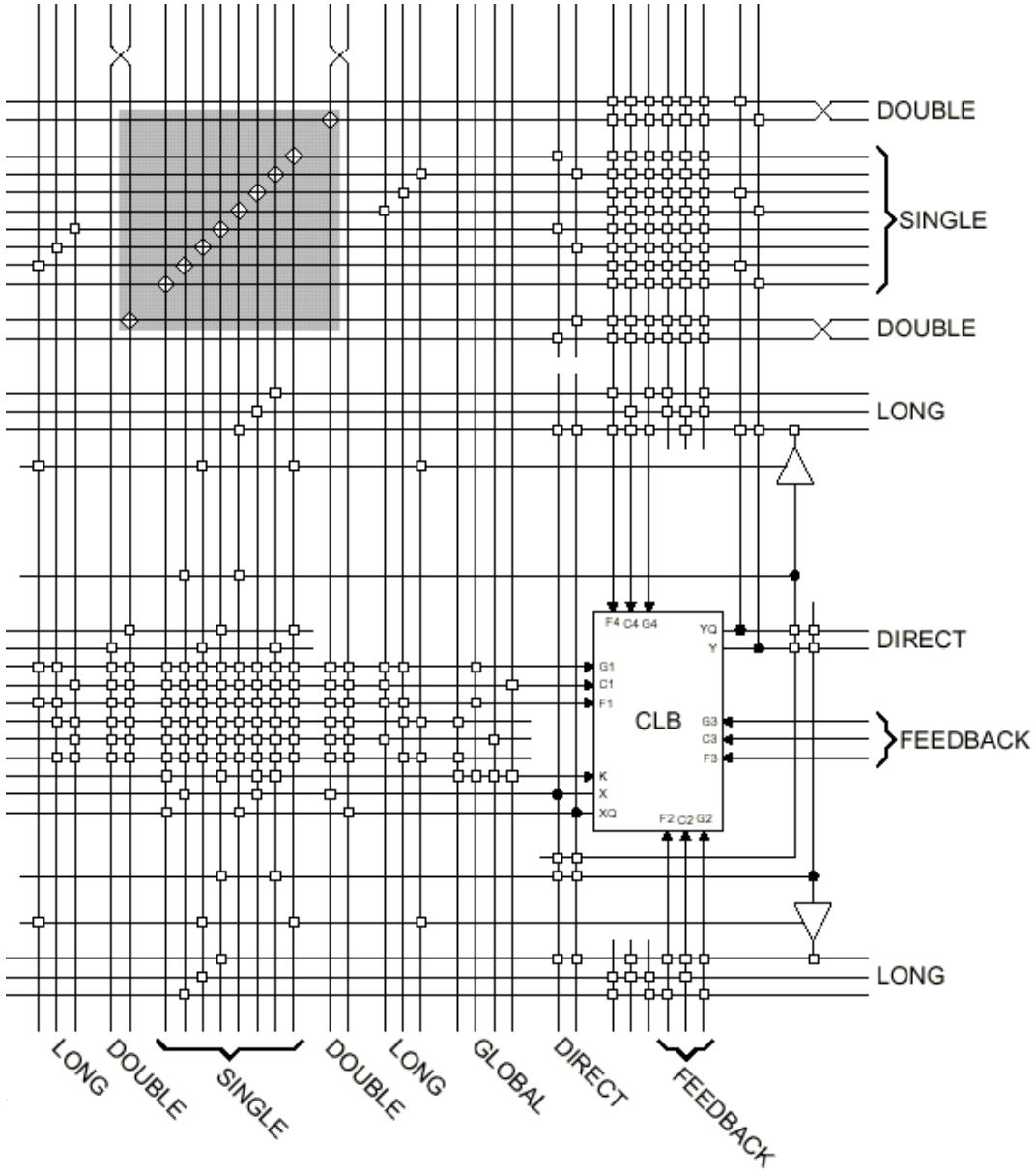


Xilinx 4000 Verbindungen

- Punkt-zu-Punkt Verbindungen (mit Abzweigungen) werden durch regelmäßige Anordnung aus 'Programmable Switch Matrix' – Elementen (PSM) erzeugt.
- Verbindungen gehen zu den Nachbarn (‚singles') und den übernächsten Nachbarn (‚doubles')



Anbindung CLB, weitere Verbindungen



- Außer PSM gibt es

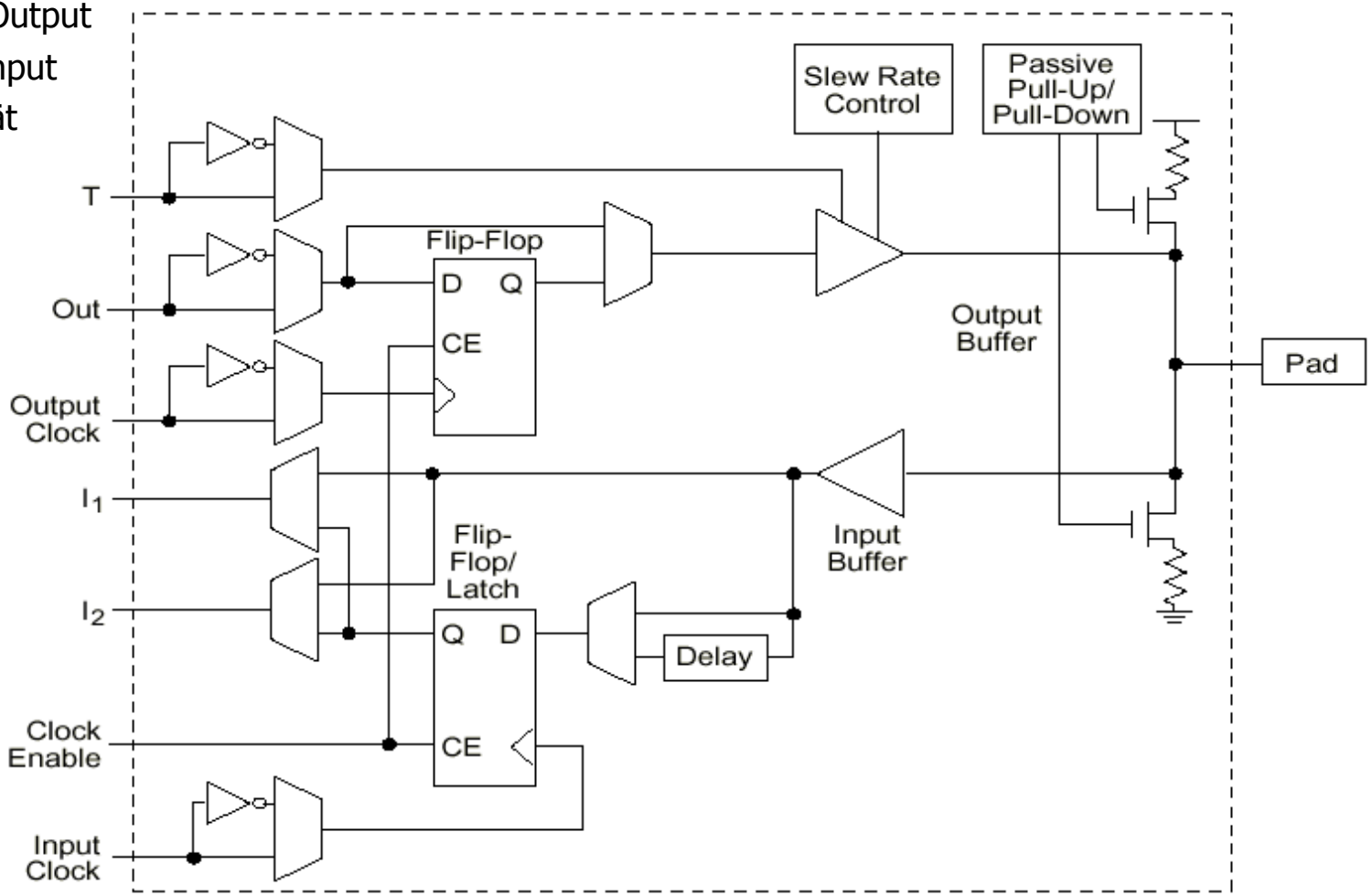
- **Long Lines** über den gesamten Chip für Busse und weite Wege
- Mehrerer **globale Taktnetze** (mit kleinem Skew)
- Die **Carry-Chain** Signale

- In der XC4000X Familie:
- **Quad-Lines**
- **Direkte Verbindungen** zwischen CLBs
- ....

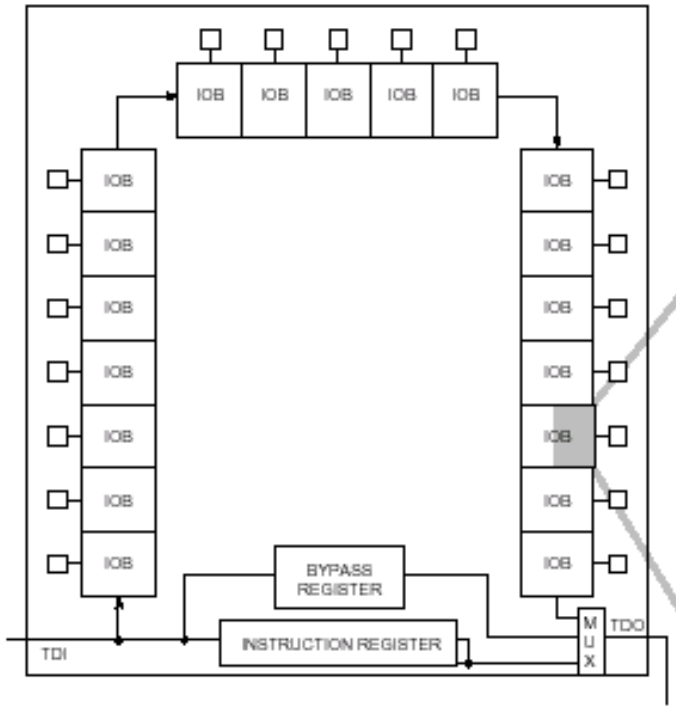
XC4000 IO-Block

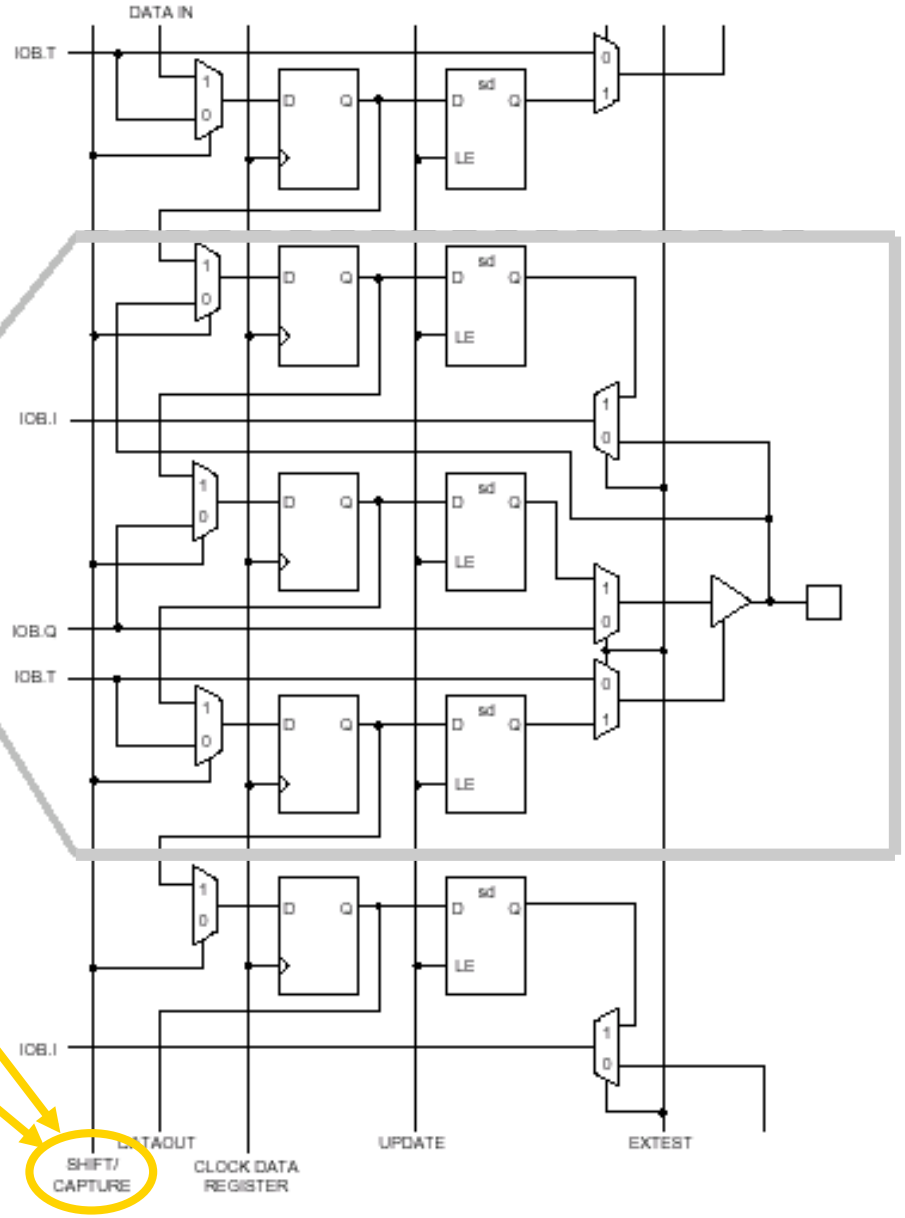
- IO-Blöcke (IOB) mit

- direct oder registered Output
- direct und registered Input
- Je mit variabler Polarität
- Output Enable
- Pullup/Pulldown
- Slew Rate control



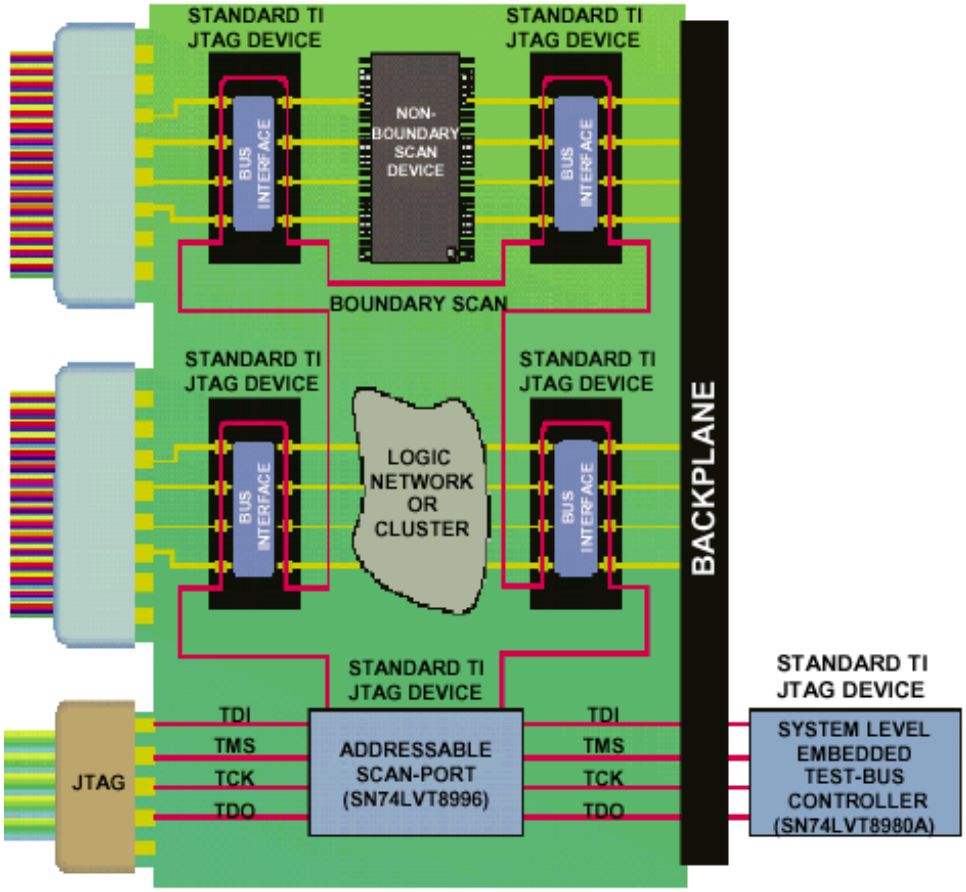
Boundary Scan (JTAG)





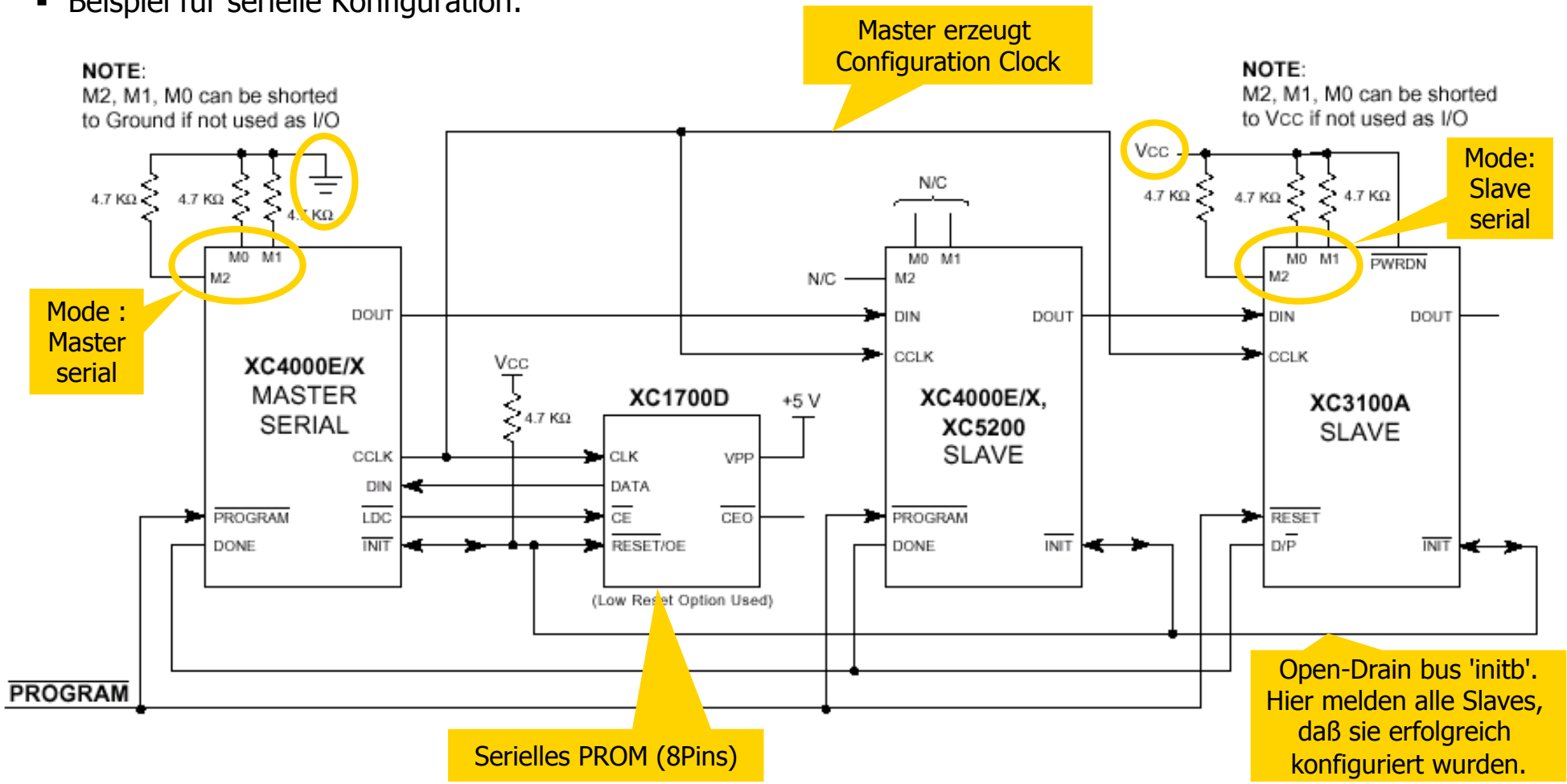
- 3 Scan FFs und update Latches pro IO Pad
- Im SHIFT-Mode als Schieberegister geschaltet
- Im CAPTURE-Mode werden PAD, Q, T ins FF übertragen
- Im EXTTEST-Mode werden I und PAD mit Werten aus dem Schieberegister belegt.

JTAG Board


STANDARD TI
JTAG DEVICE
STANDARD TI
JTAG DEVICE
BUS
INTERFACE
NON-
BOUNDARY
SCAN
DEVICE
BUS
INTERFACE
BOUNDARY SCAN
STANDARD TI
JTAG DEVICE
STANDARD TI
JTAG DEVICE
BUS
INTERFACE
LOGIC
NETWORK
OR
CLUSTER
BUS
INTERFACE
BACKPLANE
STANDARD TI
JTAG DEVICE
STANDARD TI
JTAG DEVICE
TDI
TMS
TCK
TDO
ADDRESSABLE
SCAN-PORT
(SN74LVT8996)
TDI
TMS
TCK
TDO
SYSTEM LEVEL
EMBEDDED
TEST-BUS
CONTROLLER
(SN74LVT8980A)
JTAG

Konfiguration

- XC4000 wird mit RAM konfiguriert. Daten müssen daher nach dem Einschalten erst eingelesen werden.
- Mehrere Möglichkeiten (Auswahl mit 3 'Mode'-Pins):
  - **Seriell** oder 8 Bit **Parallel**
  - **Master** (Takt wird vom Device erzeugt) oder **Slave** (Takt wird an CCLK von außen vorgegeben)
- Beispiel für serielle Konfiguration:



XC4000 Familie

Table 1: XC4000E and XC4000X Series Field Programmable Gate Arrays

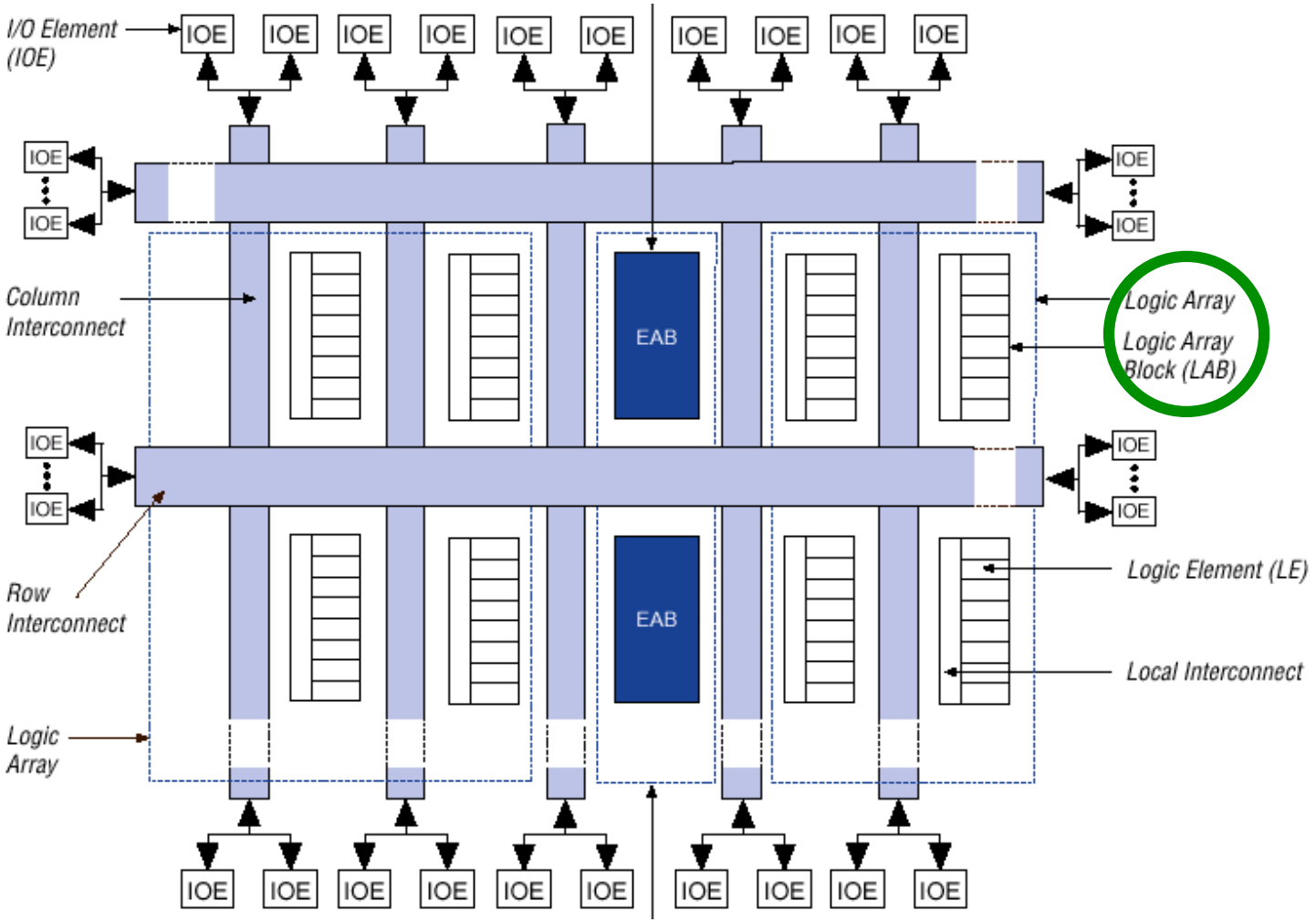
Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4002XL	152	1,600	2,048	1,000 - 3,000	8 x 8	64	256	64
XC4003E	238	3,000	3,200	2,000 - 5,000	10 x 10	100	360	80
XC4005E/XL	466	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112
XC4006E	608	6,000	8,192	4,000 - 12,000	16 x 16	256	768	128
XC4008E	770	8,000	10,368	6,000 - 15,000	18 x 18	324	936	144
XC4010E/XL	950	10,000	12,800	7,000 - 20,000	20 x 20	400	1,120	160
XC4013E/XL	1368	13,000	18,432	10,000 - 30,000	24 x 24	576	1,536	192
XC4020E/XL	1862	20,000	25,088	13,000 - 40,000	28 x 28	784	2,016	224
XC4025E	2432	25,000	32,768	15,000 - 45,000	32 x 32	1,024	2,560	256
XC4028EX/XL	2432	28,000	32,768	18,000 - 50,000	32 x 32	1,024	2,560	256
XC4036EX/XL	3078	36,000	41,472	22,000 - 65,000	36 x 36	1,296	3,168	288
XC4044XL	3800	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320
XC4052XL	4598	52,000	61,952	33,000 - 100,000	44 x 44	1,936	4,576	352
XC4062XL	5472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384
XC4085XL	7448	85,000	100,352	55,000 - 180,000	56 x 56	3,136	7,168	448

Table 20: XC4000E Program Data

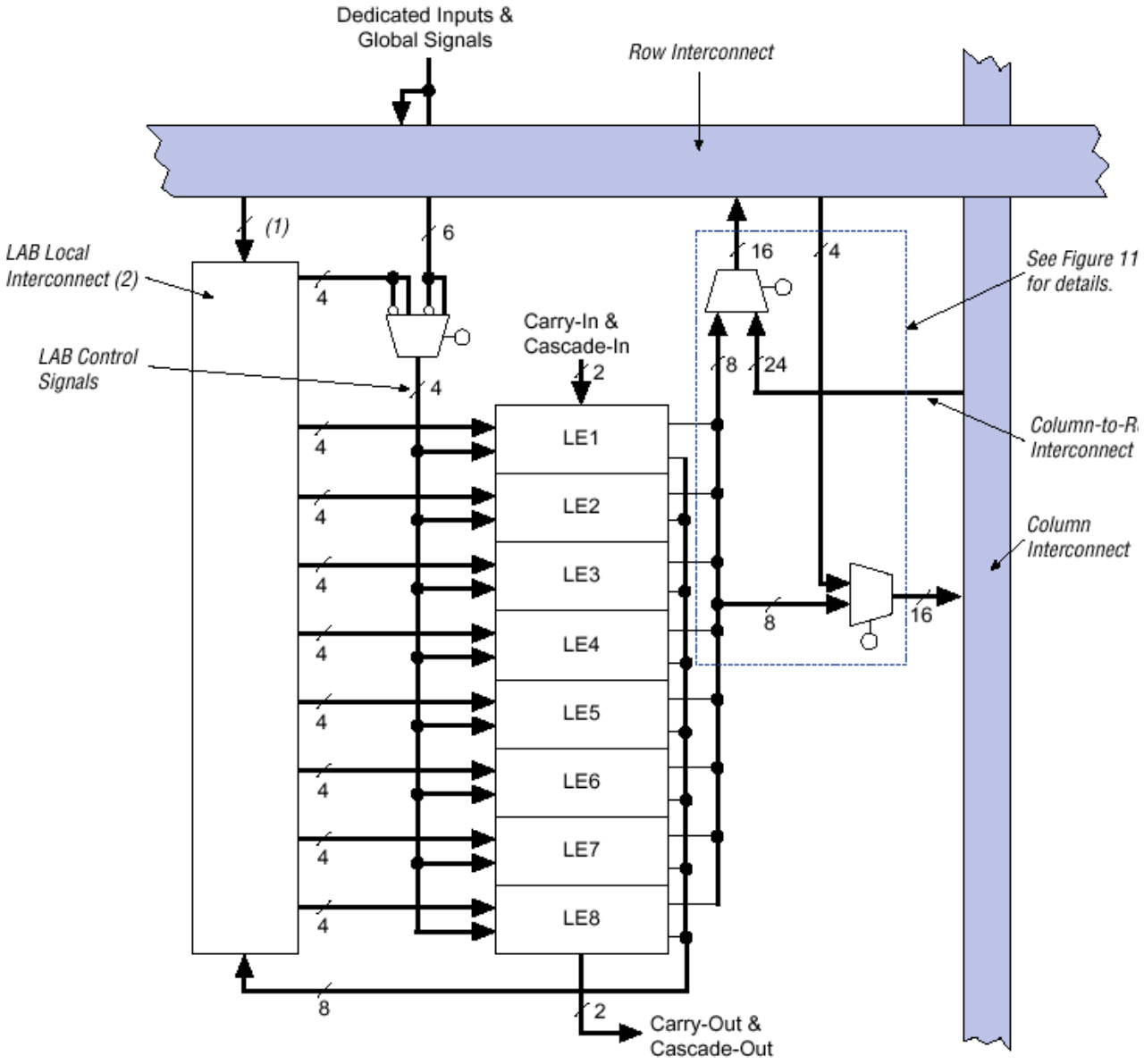
Device	XC4003E	XC4005E	XC4006E	XC4008E	XC4010E	XC4013E	XC4020E	XC4025E
Max Logic Gates	3,000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLBs (Row x Col.)	100 (10 x 10)	196 (14 x 14)	256 (16 x 16)	324 (18 x 18)	400 (20 x 20)	576 (24 x 24)	784 (28 x 28)	1,024 (32 x 32)
IOBs	80	112	128	144	160	192	224	256
Flip-Flops	360	616	768	936	1,120	1,536	2,016	2,560
Bits per Frame	126	166	186	206	226	266	306	346
Frames	428	572	644	716	788	932	1,076	1,220
Program Data	53,936	94,960	119,792	147,504	178,096	247,920	329,264	422,128
PROM Size (bits)	53,984	95,008	119,840	147,552	178,144	247,968	329,312	422,176

Altera Flex

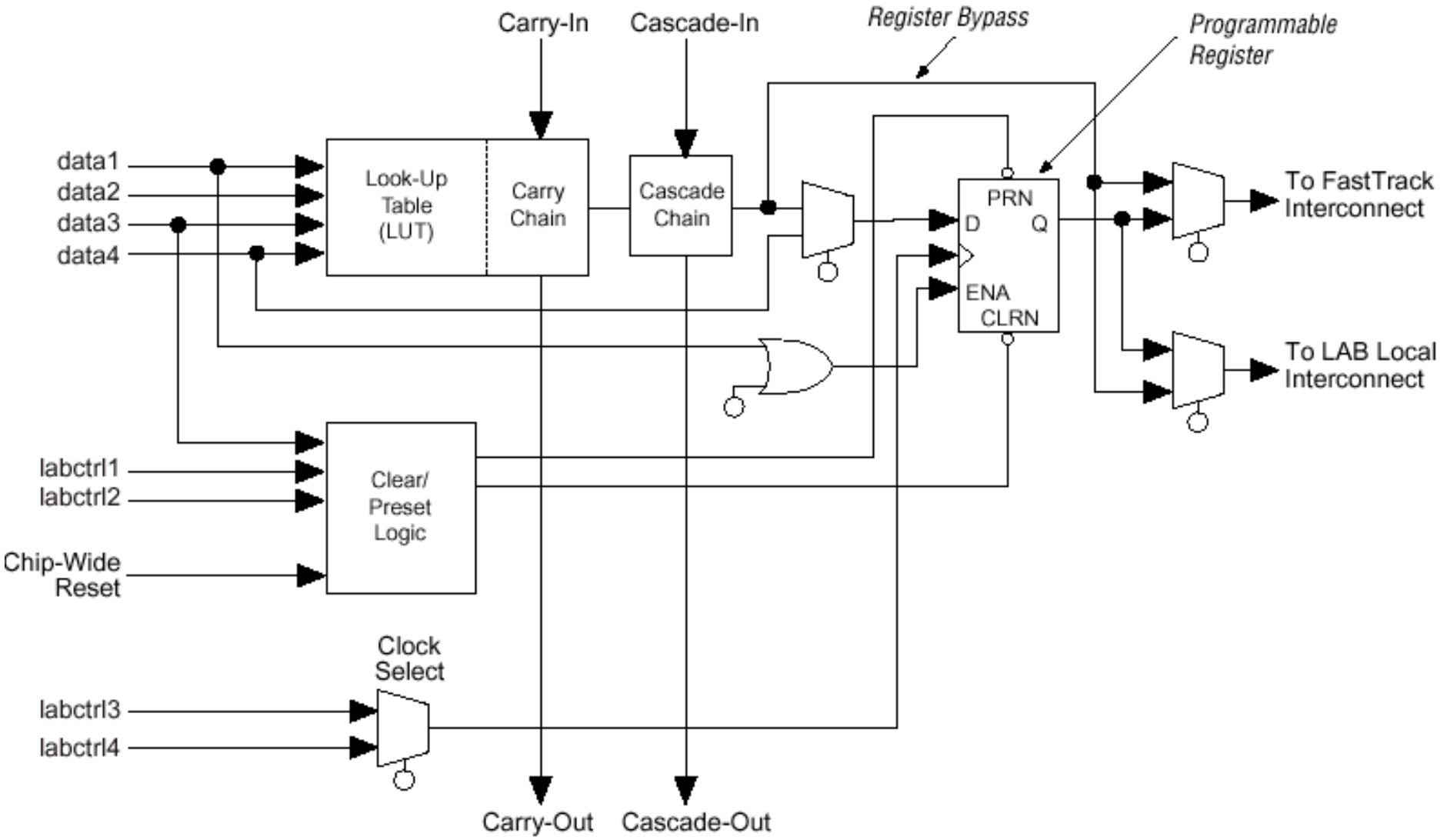
- RAM basiert, LCA mit hierarchischer Struktur: 144 'Logic Array Blöcke' (LABs) mit je 8 'Logic Elements' LE
- Dazu 6 x RAM in 'Extended Array Blöcken' EAB (256 x 8 ... 2048 x 1)



Altera Flex: Logic Elements im LAB



Altera Flex: Logic Element LE



Kommentare

- Komplexe log. Funktionen müssen in LCAs auf viele CLBs aufgeteilt werden. Dies ist eine schwierige Aufgabe für die CAD Tools!
- Die Plazierung der CLBs ist sehr wichtig um kurze Delays und gute Ausnutzung der Routing Resourcen zu bekommen. Schwieriger als bei CPLDs!
- Die Bibliotheken bieten daher neben '**Soft-Macros**' (i.W. logische Gleichungen) auch '**Hard Macros**', die für kritische Blöcke (Addierer, Zähler) die (relative) Plazierung und Verdrahtung schon beinhalten
- 'Fortgeschrittene' Benutzer können im Schaltplan schon Vorgaben über die Plazierung machen, oder festlegen, welche Routing-Resource verwendet werden soll (über ‚Property'-Attribute an Netzen und Instanzen).

- Die Hersteller bieten Bibliotheken mit komplexen Schaltungsblöcken ('cores') an, z.T. kostenlos (Werbung!, macht Bauteile attraktiv!), z.T. kostenpflichtig:
  - PCI Interface
  - Serielle, Parallele Schnittstellen
  - USB Core
  - Prozessoren (Altera: 'NIOS' embedded processor)
  - DSPs

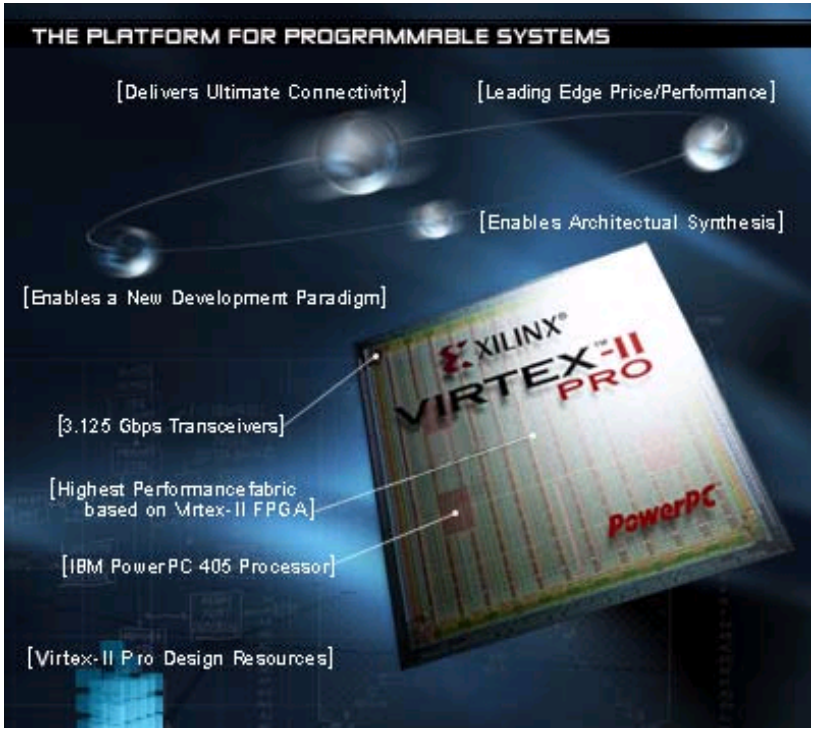
- Der Schutz von 'interlectual Property' ist wichtig. Die Firmware der Bausteine kann daher vom Benutzer geschützt werden, so daß sie von der Konkurrenz nicht ausgelesen werden kann...

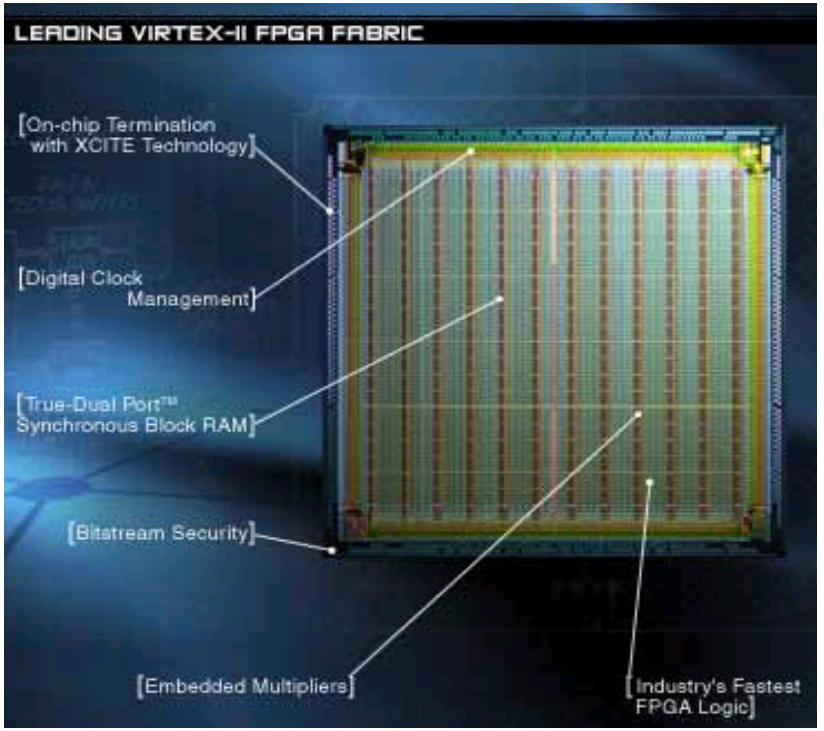
State of the Art

Erweiterungen

- Moderne Familien bieten oft noch

- Vielfältige **Routing** Resourcen
- Mehrere Taktnetze, **Clock managment Units (PLLs)**
- Spezielle **Funktionsblöcke** (Multiplizierer, CAM, ...)
- Eine Vielzahl von **IO Formaten** (LVDS, GTL, PECL, ...)
- Blöcke zur **seriellen Datenübertragung** (incl. Checksumme etc.), mehrere GHz !!!
- Eingebettete **Prozessoren** !
- Beispiel: Virtex Familie von Xilinx (inzwischen: Virtex 7)





Xilinx: Virtex-II Pro

Summary of Virtex-II Pro Features

Bis 3.125 Gbps

- High-Performance Platform FPGA Solution, Including
  - Up to twenty-four RocketIO™ embedded multi-gigabit transceivers
  - Up to four IBM® PowerPC® RISC processor blocks
- Based on Virtex™-II Platform FPGA Technology
  - Flexible logic resources
  - SRAM-based in-system configuration
  - Active Interconnect technology
  - SelectRAM™+ memory hierarchy
  - Dedicated 18-bit x 18-bit multiplier blocks
  - High-performance clock management circuitry
  - SelectI/O™-Ultra technology
  - XCITE Digitally Controlled Impedance (DCI) I/O

PLLs

Virtex-II Pro family members and resources are shown in Table 1.

Preise für Einzelstücke (Januar 2003)

Table 1: **Virtex-II Pro FPGA Family Members**

Device	RocketIO Transceiver Blocks	PowerPC Processor Blocks	Logic Cells(1)	CLB (1 = 4 slices = max 128 bits)		18 X 18 Bit Multiplier Blocks	Block SelectRAM+		DCMs	Maximum User I/O Pads	
				Slices	Max Distr RAM (Kb)		18 Kb Blocks	Max Block RAM (Kb)			
XC2VP2	4	0	3,168	1,408	44	12	12	216	4	204	$100
XC2VP4	4	1	6,768	3,008	94	28	28	504	4	348	$180
XC2VP7	8	1	11,088	4,928	154	44	44	792	4	396	
XC2VP20	8	2	20,880	9,280	290	88	88	1,584	8	564	
XC2VP30	8	2	30,816	13,696	428	136	136	2,448	8	644	
XC2VP40	0(2) or 12	2	43,632	19,392	606	192	192	3,456	8	804	
XC2VP50	0(2) or 16	2	53,136	23,616	738	232	232	4,176	8	852	$2000
XC2VP70	16 or 20	2	74,448	33,088	1,034	328	328	5,904	8	996	
XC2VP100	0(2) or 20	2	99,216	44,096	1,378	444	444	7,992	12	1,164	
XC2VP125	0(2), 20, or 24	4	125,136	55,616	1,738	556	556	10,008	12	1,200	

Neuere Xilinx Typen

- **Virtex 5**
- Seit Mai 2006
- 65 nm Technologie, 1V Core Spannung, 12 Metall-Lagen. 3.3V IO Support
- ‚LX'-Familie (ohne schnelle serielle Links) wird schon ausgeliefert
  Andere Familien folgen zu Zeit
- Neu:
  - LUTs mit 6 echten Eingängen (anstelle 2 x 4 Eingänge bisher)
    (Altera hat bei Stratix-II bereits ‚Adaptive Logic Modules')
  - 1.25 Gbps LVDS IO Zellen
  - Besseres Clock Managment (PLL, …)
  - ‚DSP Slices' für Support von single precision float operations
  - Konfigurations Bitstream mit 256 Bit verschlüsselt

- **Virtex 7**
  - Seit ~2010

- **Kintex 7**
  - Neue ‚Mid-Range' Familie

High End Modell von Altera: Stratix

Features

The Stratix family offers the following features:

Viele LEs, viel RAM

- 10,570 to 114,140 LEs; see Table 1–1.
- Up to 10,118,016 RAM bits (1,264,752 bytes) available without reducing logic resources

Dual ported RAM

- TriMatrix™ memory consisting of three RAM block sizes to implement true dual-port memory and first-in first-out (FIFO) buffers

Multiplizierer

- High-speed DSP blocks provide dedicated implementation of multipliers (at up to 250 MHz), multiply-accumulate functions, and finite impulse response (FIR) filters

PLLs

- Up to 16 global clocks with 22 clocking resources per device region
- Up to 12 PLLs (four enhanced PLLs and eight fast PLLs) per device provide spread spectrum, programmable bandwidth, clock switch-over, real-time PLL reconfiguration, and advanced multiplication and phase shifting

Viele IO Formate

- Support for numerous single-ended and differential I/O standards
- High-speed differential I/O support on up to 116 channels with up to 80 channels optimized for 840 megabits per second (Mbps)
- Support for high-speed networking and communications bus standards including RapidIO, UTOPIA IV, CSIX, HyperTransport™ technology, 10G Ethernet XSBI, SPI-4 Phase 2 (POS-PHY Level 4), and SFI-4

Intelligente Leitungs-terminierung

- Terminator™ technology provides on-chip termination for differential and single-ended I/O pins with impedance matching

Memory Interfaces

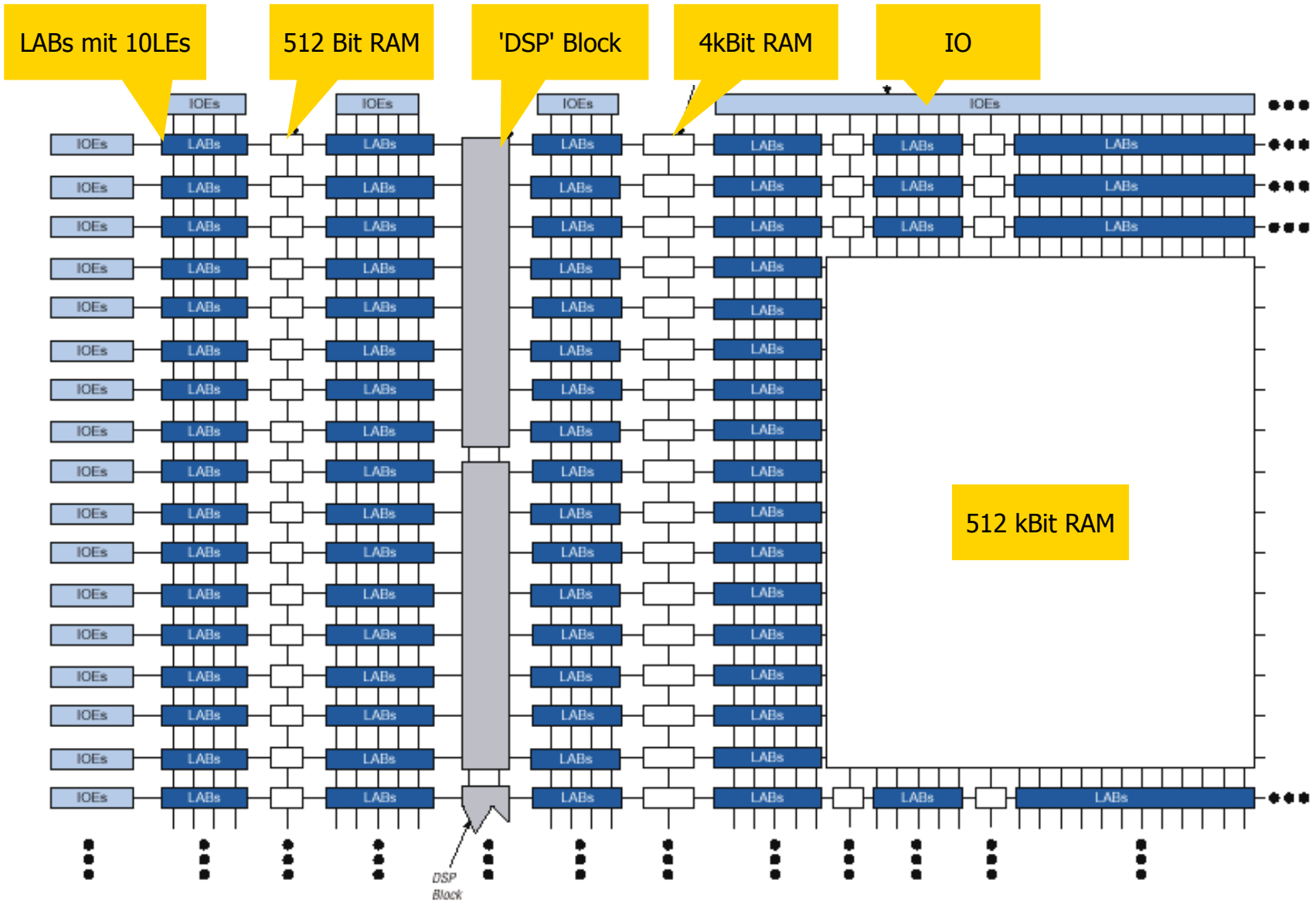
- Support for high-speed external memory, including zero bus turnaround (ZBT) SRAM, quad data rate (QDR and QDRII) SRAM, double data rate (DDR) SDRAM, DDR fast cycle RAM (FCRAM), and single data rate (SDR) SDRAM

Altera Stratix High End Parts

Feature	EP1S10	EP1S20	EP1S25	EP1S30
LEs	10,570	18,460	25,660	32,470
M512 RAM blocks (32 × 18 bits)	94	194	224	295
M4K RAM blocks (128 × 36 bits)	60	82	138	171
M-RAM blocks (4K × 144 bits)	1	2	2	4
Total RAM bits	920,448	1,669,248	1,944,576	3,317,184
DSP blocks	6	10	10	12
Embedded multipliers *(1)*	48	80	80	96
PLLs	6	6	6	10
Maximum user I/O pins	426	586	706	726

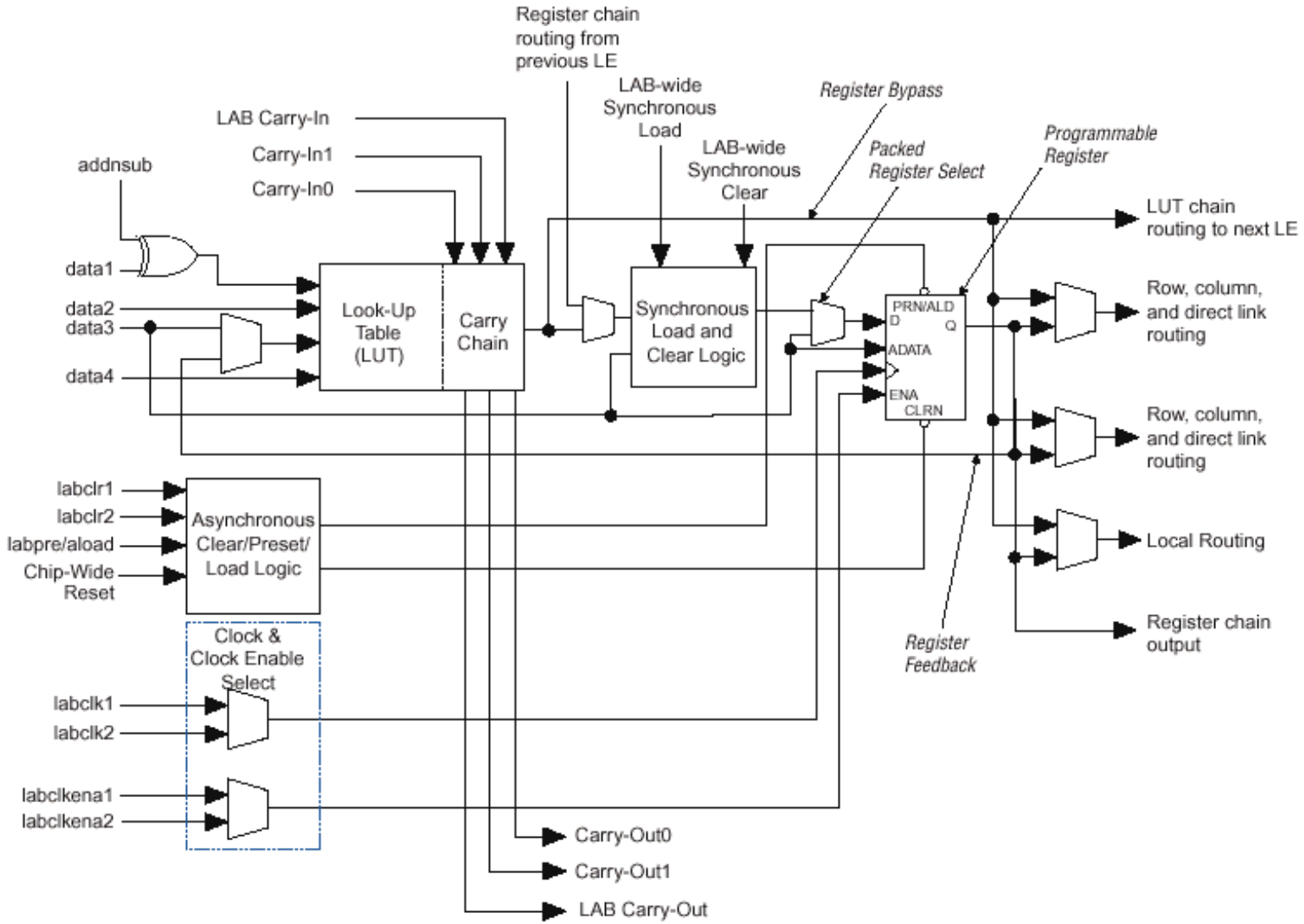
Feature	EP1S40	EP1S60	EP1S80
LEs	41,250	57,120	79,040
M512 RAM blocks (32 × 18 bits)	384	574	767
M4K RAM blocks (128 × 36 bits)	183	292	364
M-RAM blocks (4K × 144 bits)	4	6	9
Total RAM bits	3,423,744	5,215,104	7,427,520
DSP blocks	14	18	22
Embedded multipliers *(1)*	112	144	176
PLLs	12	12	12
Maximum user I/O pins	822	1,022	1,238

Stratix Elemente



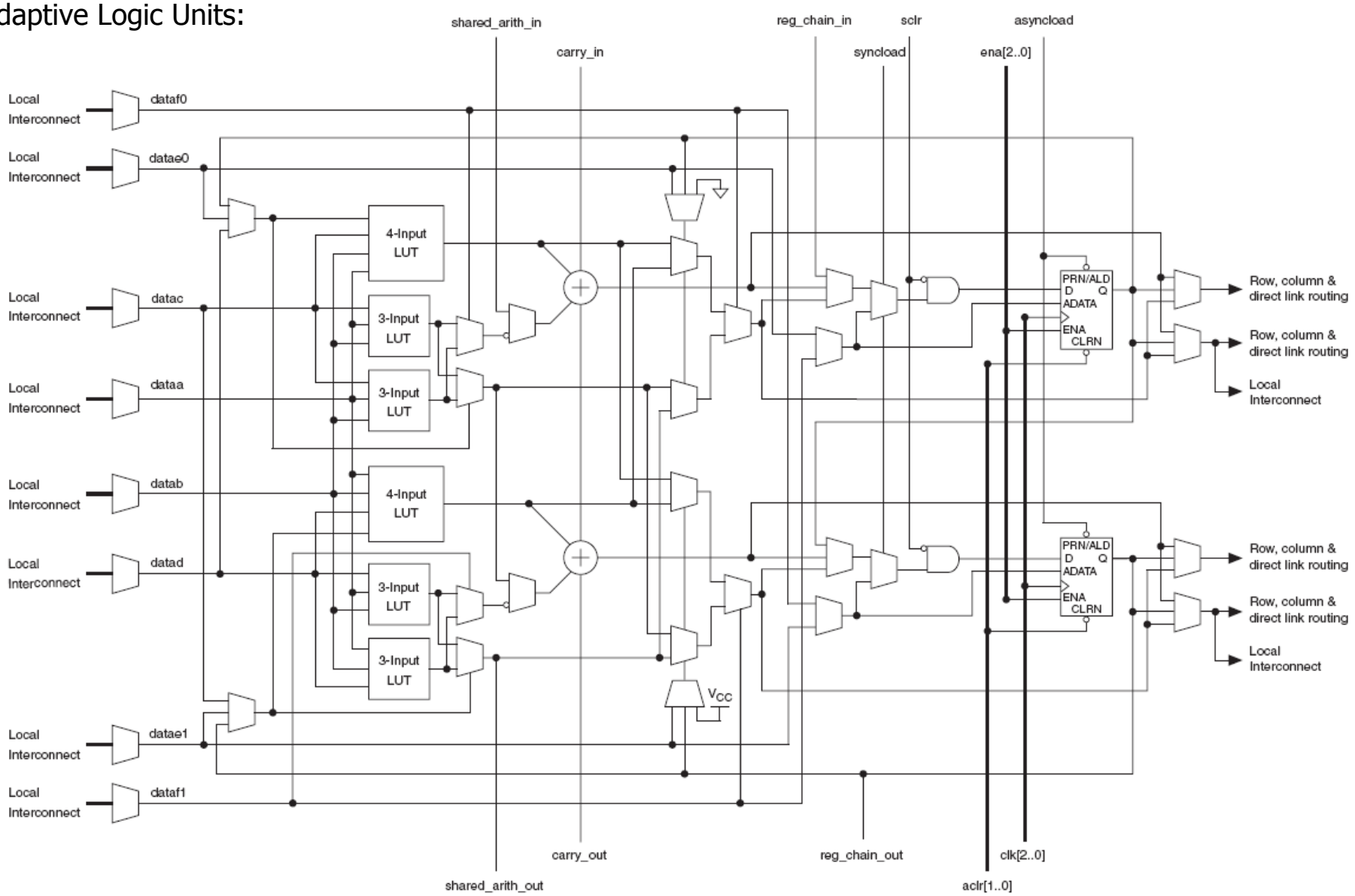
Stratix Logic Element (LE)

Figure 2–5. Stratix LE



Stratix II Familie

- Adaptive Logic Units:



Stratix II GX Familie

- GX Familie mit schnellen seriellen Links bis 6.375 GBit/s

Traumhaftes ‚Augendiagramm' bei 6.375 GBits/s (aus der Altera Werbung!)

- Inzwischen (2011): Stratix V

Hersteller & Familien

▪ Xilinx:	**Coolrunner-II**	CPLD, ursprünglich Philips (bis 1999), low power
	Artix	Low Cost
	Spartan	Nachfolger von XC4000, mit BlockRAM, ‚low cost'
	Spartan IIE	Nachfolger von Spartan, **low cost** - **auf UXIBO**
	Spartan-3	90nm Technologie, bis zu 5 Mgates - **auf Nexys-II Board**
	Spartan-6	Letztes Mitglied der Spartan Familie
	Virtex	High End Familie
	VirtexII-Pro	High End, Sehr schnelle Links, ...
	Virtex4-7	Aufeinanderfolgende, immer leistungsfähigere Familien ‚highest BW & Perf.'
	Kintex 7	Mid Range ‚**best price-performance**'
	Zynq	Mit festverdrahteten Prozessoren
▪ Altera:	**MAX-II**	'kleine' CPLD, EEPROM
	Cyclone	preiswert, ('1.5$ pro 1000LEs') weniger RAM als Stratix, **low cost**
	Cyclone IV	letztes Familienmitglied
	Aria II-V	**Mid Range**
	Stratix V	**High End**
Lattice:	**LatticeSC/XP**	Neueste Familie. Serial Links bis 3.5Gbps, PLL, DLL, ...
	LatticeECP3	
	MachX02	EPROM
▪ QuickLogic:	**ArcticLink**	Speziell mit vielen Kommunikationsmöglichkeiten

Welche Implementierung ?

- Es gibt viele Möglichkeiten, digitale Schaltungen zu implementieren:

- Diskreter Aufbau - nur für einfache Funktionen. Evtl. mit PALs
- CPLD - für mittlere Komplexitäten. 'garantierte' Geschwindigkeit
- LCA, FPGA - hohe Flexibilität
- Gate Array - keine 'Verschwendung' von Ressourcen, Möglichkeiten eingeschränkt (RAM..)
- Full Custom Chip - Alle Möglichkeiten, viel Know-How erforderlich, hohe Anfangskosten, guter Schutz der 'Interlectual Property'

- 'Rechenintensive' Probleme natürlich auch auf Prozessoren oder DSPs!