

# Untersuchung zum Einfluß der speziellen Anforderungen dynamischer Schaltungstechnik auf den Systementwurf

A.Wassatsch, D.Timmermann  
Universität Rostock

Fachbereich Elektrotechnik und Informationstechnik  
Institut für Angewandte Mikroelektronik und Datentechnik

ITG/GI/GMM-Workshop 2000  
Methoden und Beschreibungssprachen zur Modellierung und Verifikation  
von Schaltungen und Systemen

Frankfurt/Main  
01. März 2000





# Gliederung

---

- ⇒ Motivation
- ⇒ Grundlagen
- ⇒ Designanforderungen
- ⇒ Designmethodik
- ⇒ Anwendungsbeispiele
- ⇒ Zusammenfassung



# Motivation

---

⇒ Warum dynamische Logik anstatt CMOS ?

– Geschwindigkeit:

- \* die schnellsten Mikroprozessoren verwenden dynamische Schaltungstechnik (Alpha, 1GHz Prototyp IBM )
- \* halbiertes Fan-in

– Fläche:

- \* Transistoranzahl eines n-Eingangsgatter mit Register:  
 $11 + n < 16 + 2n$  (TSPC:CMOS)
- \* aber erhöhter Aufwand beim CLK-Tree

– Leistungsverbrauch:

- \* ist nicht so abhängig von der Taktfrequenz wie CMOS
- \* Stromverbrauch durch Signalpegel und nicht durch Signalpegelwechsel

→ DFG: VIVA

– True Single Phase Clock (TSPC) als Schaltungstechnik für Standardzellen :

- \* robust
- \* digitale Beschreibung des Zell-Verhaltens möglich

TSPC aber kein Allheilmittel für jede Schaltung !



## Motivation (2.)

---

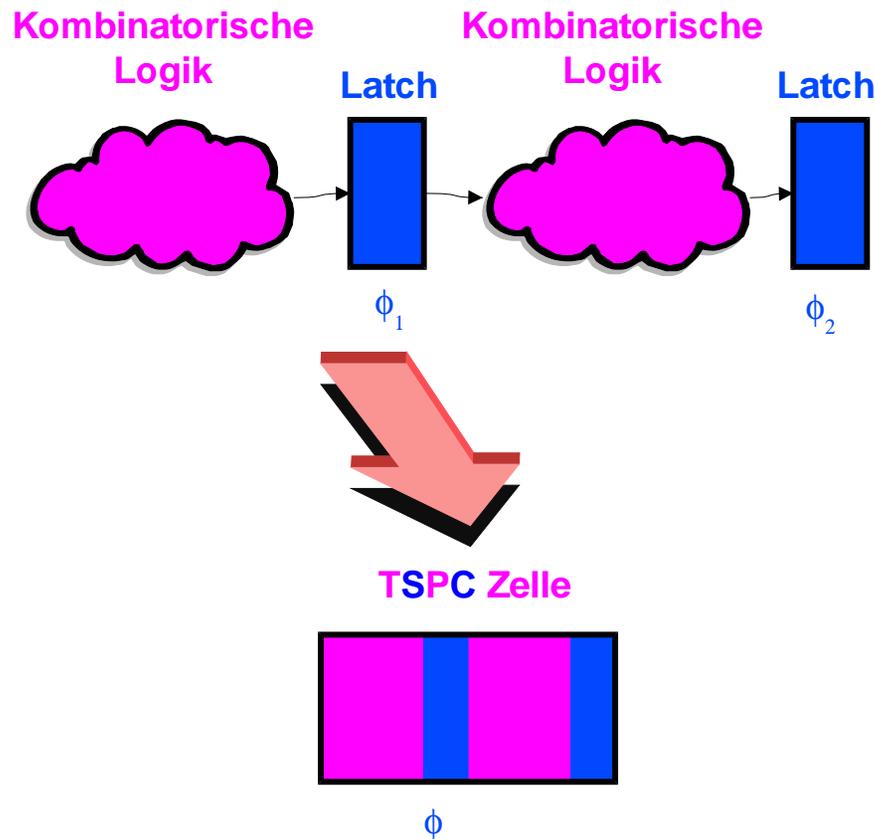
⇒ Warum Designflow erweitern ?

- dyn. Logik-Synthese von Tools nicht unterstützt, strenge Unterscheidung zwischen kombinatorischen und sequentiellen Zellen, Synthesebibliothek muß OR/AND, INV und REG beinhalten
- “Schematic entry” der Schaltungen → Handarbeit, paßt in keinen modernen Designflow
- Workaround 1: strukturelle HDL-Beschreibung, die auf die Besonderheiten von dyn. Logik eingeht → Schematic-Entry like
- Workaround 2: Trennung von Logik und Pipeline-Register-File in der HDL-Beschreibung, nach der Logik-Synthese “balance-registers” → resultierende Netzlist nicht für TSPC anwendbar

Entwicklung einer Designflow-Erweiterung notwendig !



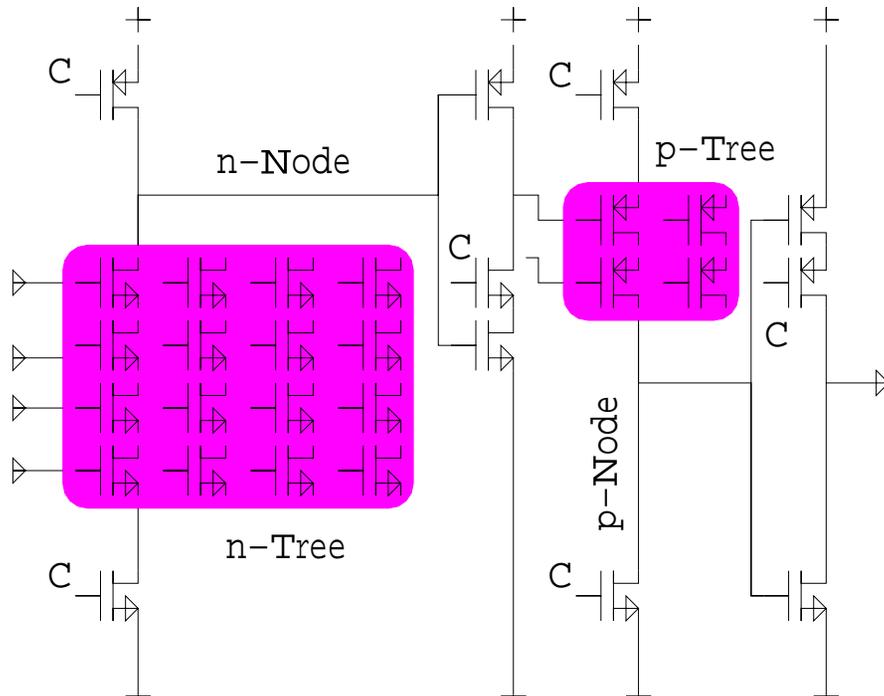
# Grundlagen



- ⇒ Verschmelzung von kombinatorischen und sequentiellen Elementen zu einer Zelle
- ⇒ keine rein kombinatorischen Blöcke möglich
- ⇒ Verarbeitungsprozeß vom Takt abhängig
- ⇒ Funktionsprinzip beruht auf kapazitiver Ladungsspeicherung
- ⇒ Logikfunktion in nur einen Transistorbaum realisiert
- ⇒ Beispiele: C<sup>2</sup>MOS TSPC, Domino-Logik, CVSL, DCVSL, DCSL



# True Single Phase Clock - Logik



## ⇒ Vorteil:

- benötigt nur ein Taktsignal
- geringeres FAN-IN durch einen Transistorbaum
- Nutzung beider Taktphasen durch alternierende Aktivierung

## ⇒ Nachteil:

- erhöhte Belastung des Taktes
- kann nur nicht-negierende Logikfunktionen darstellen

## ⇒ zeitlich getrennte Betriebszustände

- Aufladephase (precharge): Speicherung einer kleinen Ladungsmenge auf internem Knoten
- Entscheidungsphase (evaluate): logikabhängige Entladung





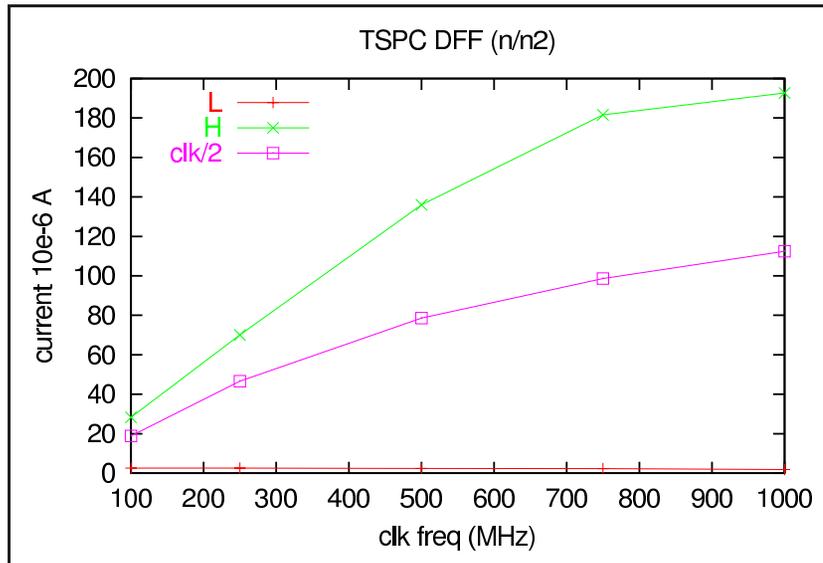
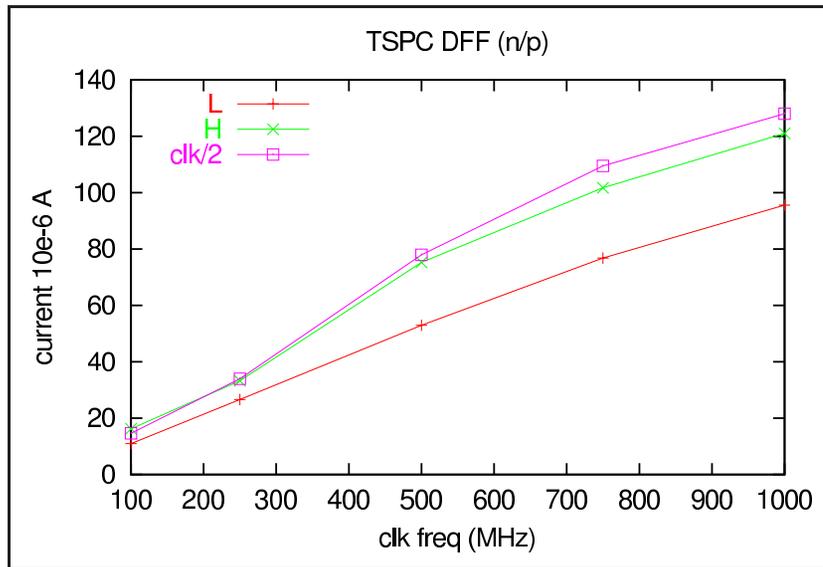
# Designanforderungen

---

- ⇒ bezüglich Latenzzeit unkritische bzw. tolerante Schaltungsumgebung
- ⇒ besonders geeignete Schaltungsarchitekturen
  - arithmetische Operatoren mit bitbreiten-unabhängiger Laufzeit (CS-, SD-Addierer)
  - pipeline-fähige Signalverarbeitung mit nichtrekursivem Datenfluß
    - \* digitale Filter
    - \* räumlich entwickelte iterative Algorithmen (CORDIC, DES)
- ⇒ ungünstige Schaltungsstrukturen
  - laufzeitbasierte Konstrukte (Monoflop, RS-Flipflop)
  - Architekturen mit stark ausgeprägter interner Serialität (Ripple Carry Addierer (RCA))
  - Rückkoppelungen im Datenfluß über mehr als eine Zellebene (Zähler)



# Minimierung des Leistungsverbrauchs



- ⇒ nicht die Veränderung eines internen Zustandes benötigt Strom, sondern die kontinuierliche Auffrischung dieses Zustandes ⇒ DRAM
- ⇒ Leistungsverbrauch ist abhängig von der Verteilung der Signalpegelhäufigkeit und **nicht** von der Häufigkeit des Signalpegelwechsels
- ⇒ Reduktion des Leistungsverbrauchs möglich
  - durch Architekturwahl
  - durch verändertes Optimierungsziel
  - durch angepaßte Zellbibliothek



# Minimierung des Leistungsverbrauch (2.)

## Einfluß der Architekturentscheidung

⇒ Auswahl von verbrauchminimalen Zustandskodierungen bei unvollständiger Coderaumbelegung

⇒ One Hot Encoding von Zustandsmaschinen

⇒ Ringzähler

Beispiel 4 Bit Vektor

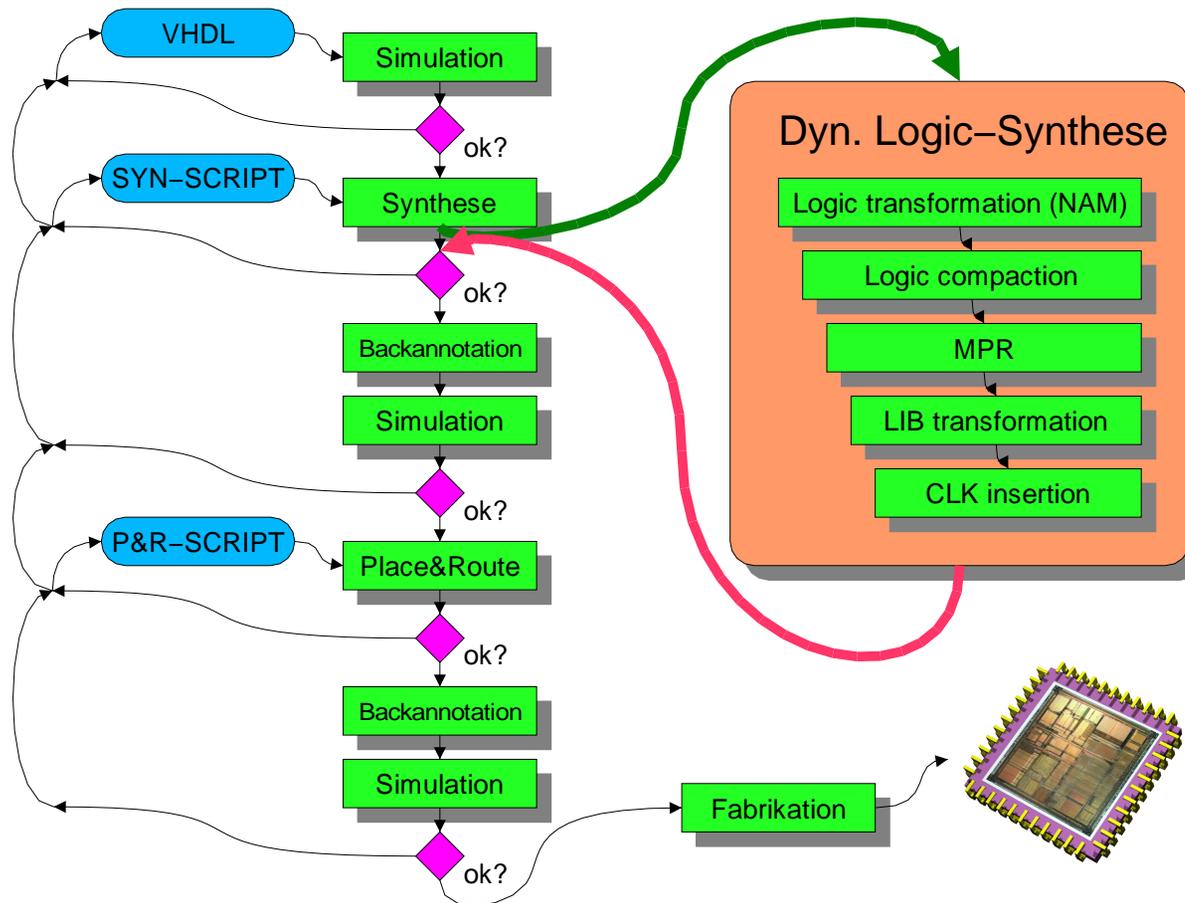
0	1	2	3	4
0000	0001	0011	0111	1111
	0010	0101	1011	
	0100	0110	1101	
	1000	1001	1110	
		1010		
		1100		

Ringzähler  $\frac{1}{m}$  aktiv  $\Rightarrow 1$

Binärzähler 0.5 aktiv  $\Rightarrow 0.5n$



# Designmethodik

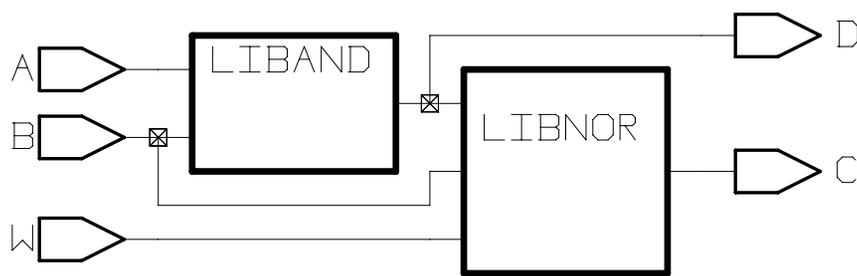


## ⇒ TSPC-Designflow

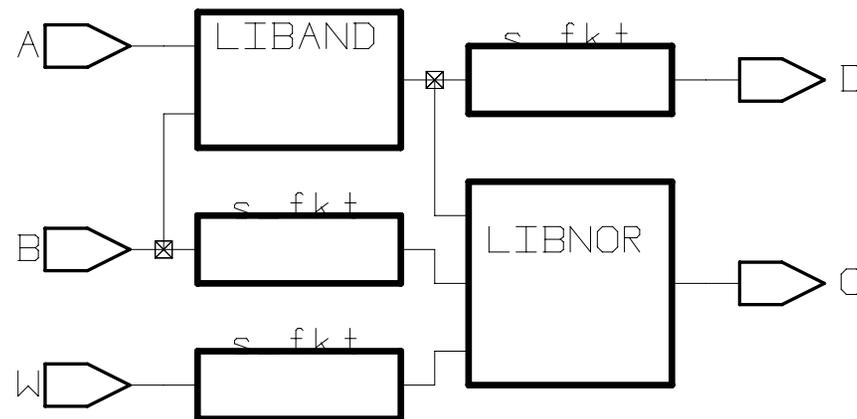
- Einbindung der dynamischen Schaltungstechnik in den Standard CMOS Designflow
- Kapselung der besonderen dynamischen Eigenschaften in Simulations- und Synthesebibliotheken
- Einsatz von Standardzellen



# Micro Pipeline Reorganizer (MPR)



Ausgangsnetzliste

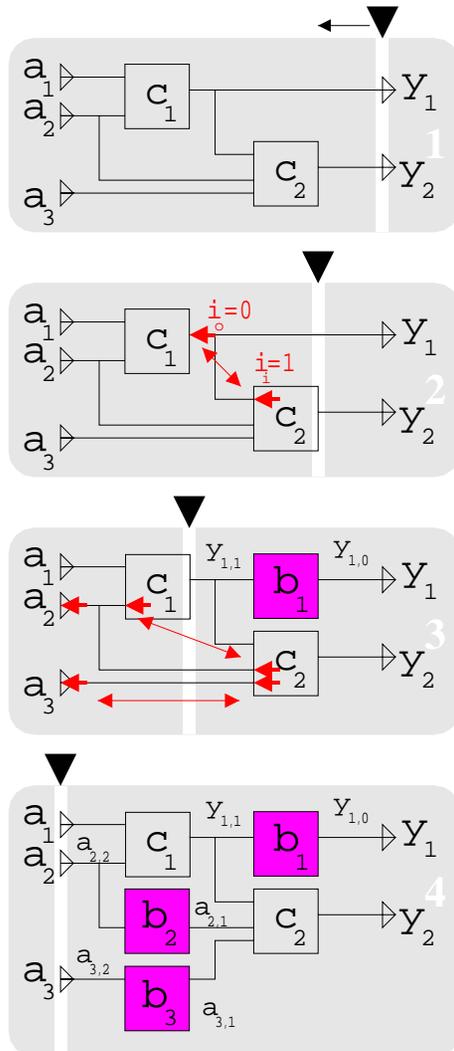


bearbeitet Netzliste

- ⇒ Ziel: Verbesserung des Durchsatzes einer gegebenen Schaltung
- ⇒ Aufgabe: Aufbau einer Pipelinestruktur durch Einfügen von zusätzlichen Register-Zellen
- ⇒ Bedingung:
  - Grundzellen mit nur einem Ausgang
  - keine Rückkopplung in der Netzliste
- ⇒ Netzlistenformate: XNF, strukt. VHDL



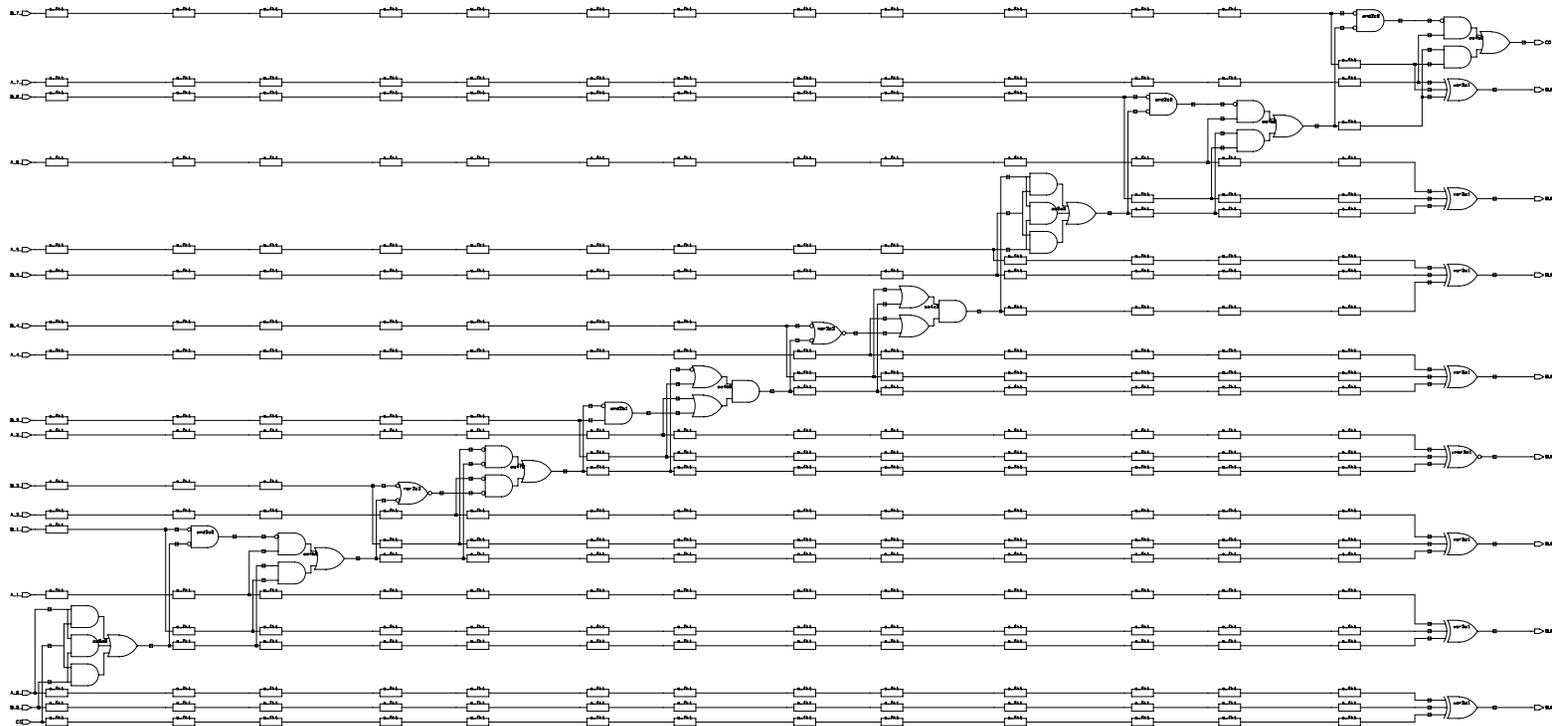
# MPR Algorithmus



- ⇒ Optimierung beginnt an den Ausgangs-ports der Netzliste mit der Markierung der Netzlevel
- ⇒ Erfassung der treibenden Zellausgänge
- ⇒ Überprüfung der Netzlevel der zugehörigen Zelleingänge
- ⇒ gegebenenfalls Einfügen von Bufferzellen (Register)
- ⇒ Erhöhung des Netzlevelvalues
- ⇒ Wiederhole solange, bis alle betrachteten Netze nur noch die Eingangsports der Netzliste als Treiber haben



# Design-Beispiele



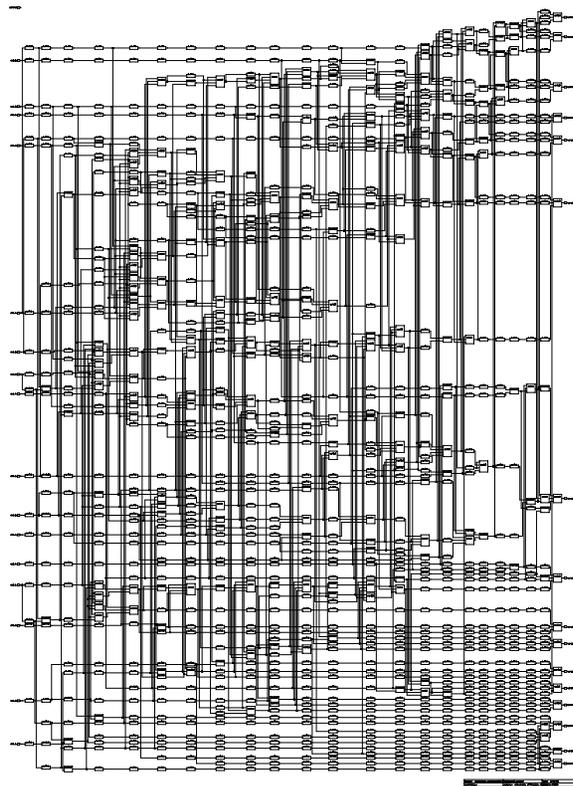
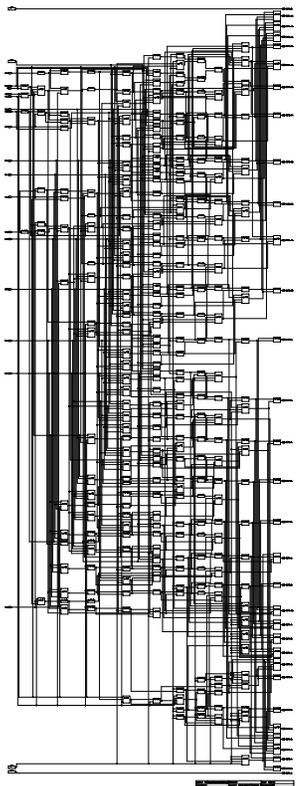
MPR 8 bit Ripple-Carry Addierer

⇒ Pipelineexpansion durch starke interne serielle Signalabhängigkeit ⇒ Faktor  $2n - 2$

⇒ hat einen Pipelineeffektivitätsgrad  $E = \frac{b}{m} \frac{\sum_{i=1}^m \frac{N_i}{n_i}}{m} = \frac{8}{14} \frac{9.62}{14} = 0.392$



# Design-Beispiel: MPR 8x8 CSA Multiplizierer

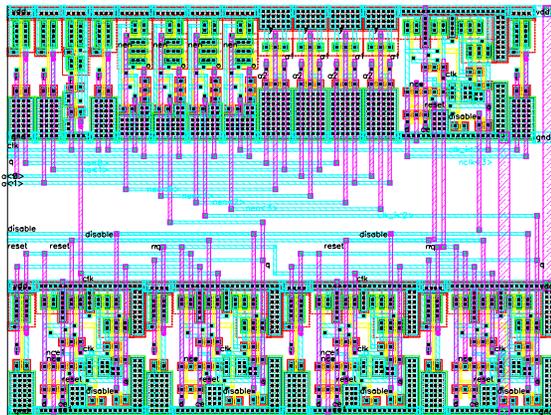
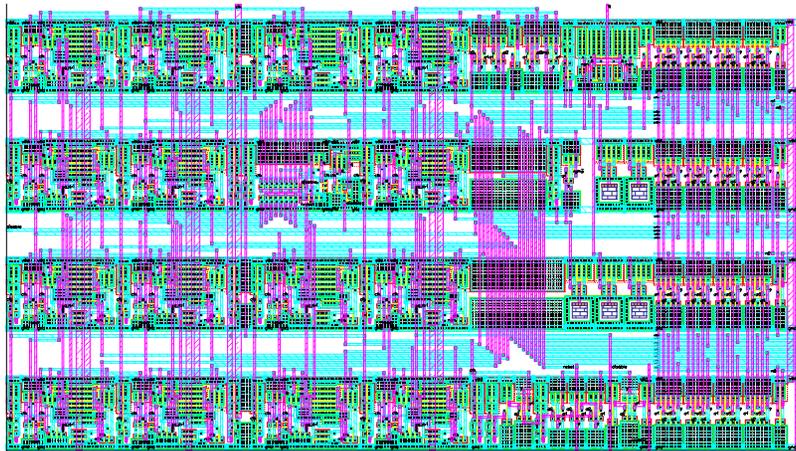


- ⇒ 8bit x 8bit Carry-Save Multiplizierer aus der Synopsys Designware
- ⇒ Erhöhung der Pipelinestufenanzahl
- ⇒ Pipelineeffektivitätsgrad

$$E = \frac{b \sum_{i=1}^m \frac{N_i}{n_i}}{m \quad m} = \frac{16 \quad 20.98}{23 \quad 23} = 0.635$$



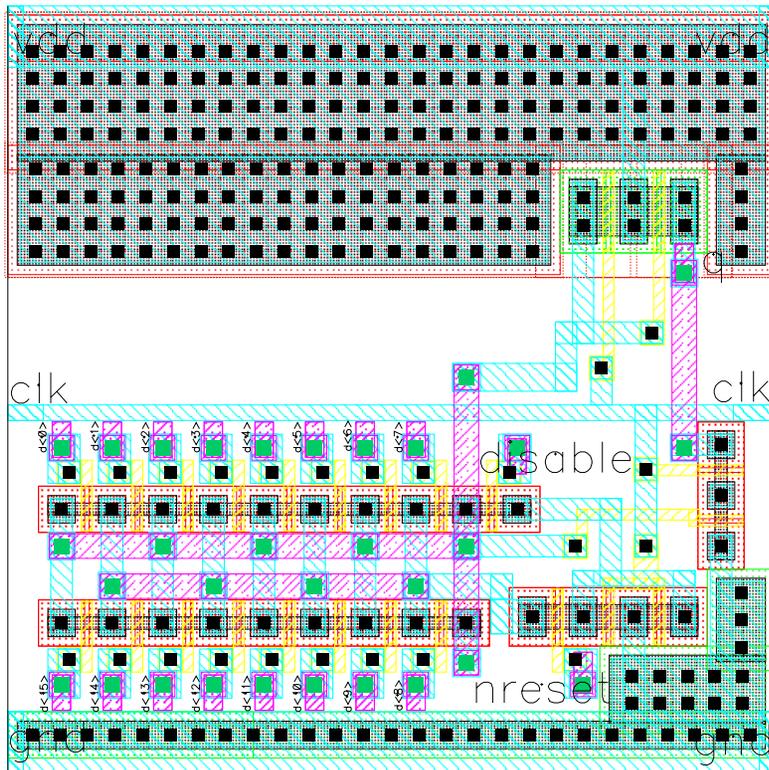
# TSPC Beispiel: PLL Frequenzteiler



- ⇒ 0.35 $\mu$  3.3V Technologie
- ⇒ Loop Teiler:
  - 650 MHz / 455 MHz low power Version
  - power-down-mode, shift-register, watch-dog
  - 1-16:1 Teiler, Fläche 270x205 $\mu m$
  - asynchrones Reset !!!
- ⇒ Board/Pad:
  - 1/4/16/32:1 Teiler
  - 714 MHz
- ⇒ CMOS für Timing unkritische Decoder-Schaltungen und Signalpufferung



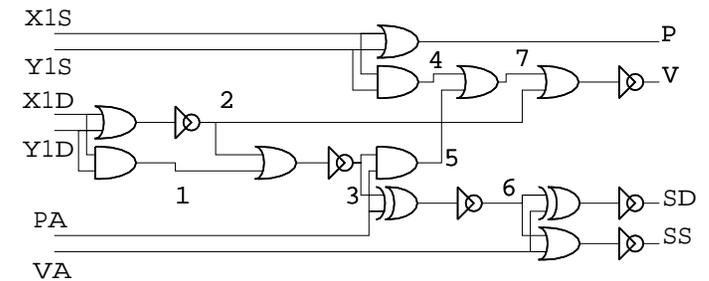
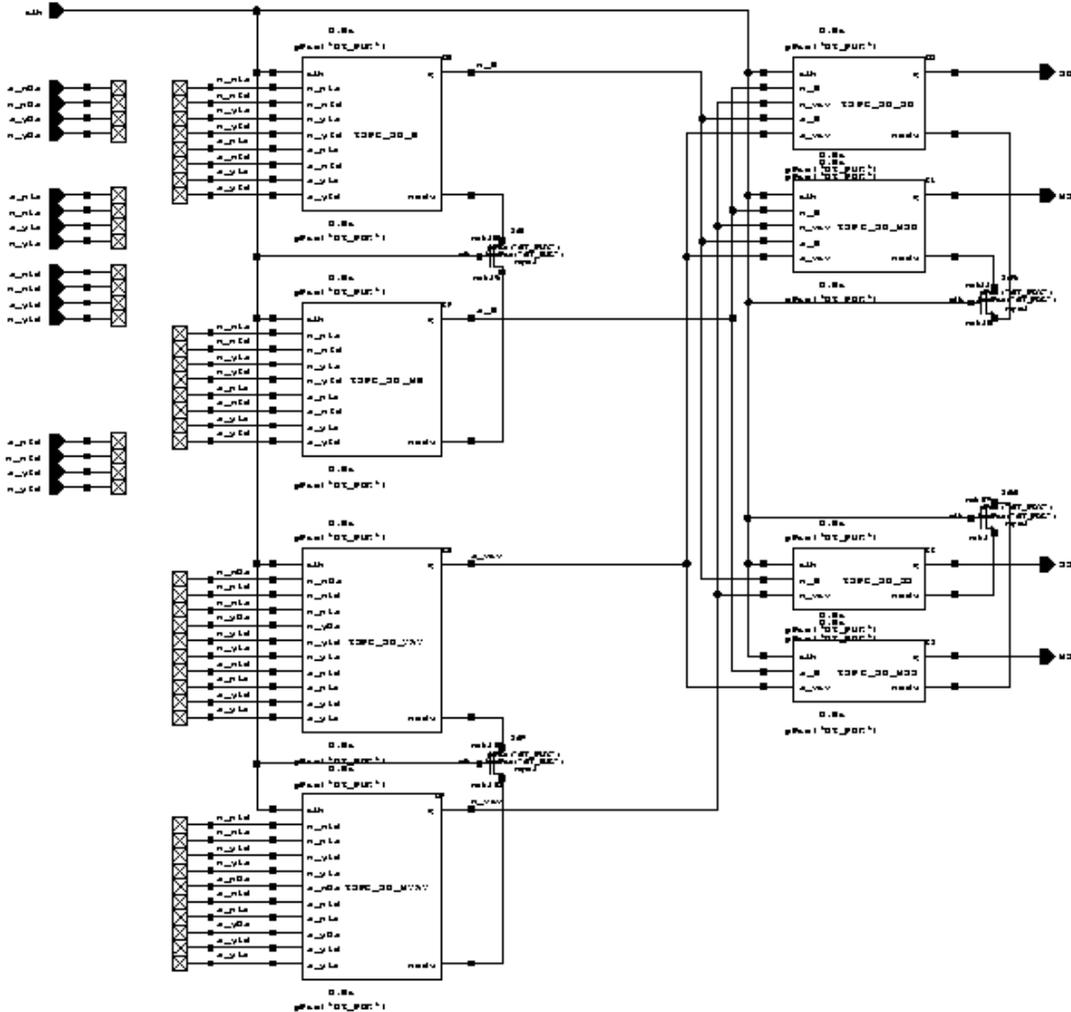
# TSPC Beispiel: Detail Loop Teiler



- ⇒ 0.35μ 3.3V Technologie
- ⇒ 16-fach TSPC-ODER Implementierung
- ⇒ in CMOS-Implementation nur gestuft möglich ( keine 16 MOS in Reihe )
- ⇒ eingesetzt im Loop-Teiler
- ⇒  $f_{max}$  von 350MHz auf 455MHz verbessert gegenüber “wired OR”-Realisierung



# DTSPC Beispiel: Signed Digit Adder Zelle



⇒ 0.6 $\mu$  5V AMS-CUB  
Technologie

⇒ SD-Addition in einem  
Takt gegenüber 8 Tak-  
ten bei einzelnen Zellen



# Alternative Einsatzmöglichkeiten

---

- ⇒ Beschleunigung von Standard CMOS Designs
  - Erhöhung der Performance durch automatisch generierte Pipeline-Struktur, Pipelinetiefe wird durch Netzliste bestimmt, keine Vorgabe durch den Entwickler wie beim “Balance-Register”-Ansatz
  - Einsatz für FPGA-Entwicklung: eine Pipelinestufe je CLB
- ⇒ Realisierung von Wavepipelines
  - Ansatz: Austausch der Registerfunktionalität durch einfache Verzögerungsglieder (Buffer)
  - bei FPGA-Entwicklung: Vorgabe der zu erreichenden Routinglaufzeiten durch feinere Granularität der Netzliste vereinfacht



# Zusammenfassung

---

- ⇒ Einsatz dynamischer Schaltungstechnik erfordert die Nutzung angepaßter Schaltungsarchitekturen
- ⇒ Integration in Standard-CMOS Designflow durch partielle Erweiterung möglich
- ⇒ Evaluierung der alternativen Einsatzmöglichkeiten des Toolsets
- ⇒ Entwicklung eines Standardzell-Library-Generators für TSPC
- ⇒ Referenzimplementierung einer TSPC-Standardzell-Library auf der Grundlage der AMS  $0.35\mu m$  bzw.  $0.6\mu m$  Technologie
- ⇒ Implementation von Referenzdesigns (CORDIC, Digit-Online-Neuro, DES )

● Gliederung . . . . .	2
● Motivation . . . . .	3
4	
● Grundlagen . . . . .	5
6 7	
● Designanforderungen . . . . .	8
9 10	
● Designmethodik . . . . .	11
12 13	
● Design-Beispiele . . . . .	14
15 16 17 18	
● Alternative Einsatzmöglichkeiten . . . . .	19
● Zusammenfassung . . . . .	20

# List of Slides

- 2 Gliederung
- 3 Motivation
- 4 Motivation (2.)
- 5 Grundlagen
- 6 True Single Phase Clock - Logik
- 7 Differential TSPC
- 8 Designanforderungen
- 9 Minimierung des Leistungsverbrauchs
- 10 Minimierung des Leistungsverbrauch (2.)
- 11 Designmethodik
- 12 Micro Pipeline Reorganizer (MPR)
- 13 MPR Algorithmus
- 14 Design-Beispiele
- 15 Design-Beispiel: MPR 8x8 CSA Multiplizierer
- 16 TSPC Beispiel: PLL Frequenzteiler
- 17 TSPC Beispiel: Detail Loop Teiler
- 18 DTSPC Beispiel: Signed Digit Adder Zelle
- 19 Alternative Einsatzmöglichkeiten
- 20 Zusammenfassung

# Literatur

- [Gronowski et al., 1998] Gronowski, P., Bowhill, W., Preston, R., Gowan, M., and Allmon, R. (1998). High-performance microprocessor design. *Journal of Solid State Circuits*, 33(5):676–686.
- [Heller et al., 1984] Heller, L., Griffin, W., Davis, J., and Thoma, N. (1984). Cascode voltage switch logic: A differential cmos logic family. In *Proceedings IEEE International Solid-State Circuits Conference*, pages 16–17.
- [Matson et al., 1998] Matson, M., Bailey, D., Bell, S., Biro, L., Butler, S., Clouser, J., Farrel, J., Gowan, M., Priore, D., and Wilcox, K. (1998). Circuit implementation of a 600mhz superscalar risc microprocessor. In *Proceedings of the International Conference on Computer Design*, pages 104–110.
- [Ng et al., 1996] Ng, P., Balsara, P., and Steiss, D. (1996). Performance of cmos differential circuits. *Journal of Solid State Circuits*, 31(56):841–846.
- [Øye, 1996] Øye, J. (1996). *A High Speed Cell Library in CMOS for Bit-Serial Implementation of DSP Algorithms*. PhD thesis, Norwegian University of Science and Technology, Dept. of Physical Electronics.
- [Pihl, 1996] Pihl, J. (1996). *Design Automation of High Speed Digital Signal Processing in VLSI with Application in Speech Recognition Systems Based on Hidden Markov Models*. PhD thesis, Norwegian University of Science and Technology, Dept. of Physical Electronics.
- [Pihl and Aas, 1995] Pihl, J. and Aas, E. (1995). Automated logic synthesis with cdpd circuit technique. In *Proceedings of the Norchip Conference*, Copenhagen.
- [Pihl et al., 1994] Pihl, J., Øye, J., and Aas, E. (1994). Semi-automatic synthesis with standard cells in cdpd circuit technique. In *Proceedings of the Norchip Conference*, Gothenburg.
- [Pihl et al., 1996] Pihl, J., Øye, J., and Aas, E. (1996). Logic synthesis with the cdpd circuit technique. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS'96)*, Atlanta.
- [Pihl et al., 1997] Pihl, J., Øye, J., and Aas, E. (1997). *Logic Synthesis with High Speed CMOS Circuit Techniques*. Kluwer Academic Publishers.
- [Posluszny et al., 1998] Posluszny, S., Aoki, N., Boerstler, D., Burns, J., Dhong, S., Ghoshal, U., Hofstee, P., LaPotin, D., Lee, K., Meltzer, D., Ngo, H., Nowka, K., Silbermann, J., Takahashi, O., and Vo, I. (1998). Design methodology for a 1.0 ghz microprocessor. In *Proceedings of the International Conference on Computer Design*, pages 17–23.

- [Sundsbo, 1997] Sundsbo, I. (1997). *Analysis and VLSI design of synthesis filter bank for image subband coding*. PhD thesis, Norwegian University of Science and Technology, Dept. of Physical Electronics.
- [Timmermann et al., 1999] Timmermann, D., F.Grassert, and Wassatsch, A. (1999). Schaltungstechnik und Architekturen für mobile digitale Signalverarbeitung mit drahtloser Kommunikation bei niedrigster Leistungsaufnahme. In *1.Kolloquium im Rahmen des Schwerpunktprogrammes "Grundlagen und Verfahren verlustleistungsarmer Informationsverarbeitung (VIVA) der DFG*.
- [Wassatsch, 1998] Wassatsch, A. (1998). Algorithmen für Umsetzung dynamischer Schaltungstechnologien. Technical report, Universität Rostock, FB ETIT, Institut MD.
- [Wassatsch et al., 1998] Wassatsch, A., Dolling, S., and Timmermann, D. (1998). Area minimization of redundant cordic pipeline architectures. In *Proceedings of the International Conference on Computer Design*, pages 136–141.
- [Yuan et al., 1987] Yuan, J., Karlsson, I., and Svensson, C. (1987). A true single phase clock dynamic cmos circuit technique. *IEEE Journal of Solid State Circuits*, SC-22:899–901.