

Verlustleistungsreduzierung bei dynamischen TSPC-Schaltungstechniken

F. Grassert, D. Timmermann



Übersicht

- Stand der Entwicklungen / Rückblick
- Neuer Ansatz RTL-Ebene
- Ergebnisse umfangreicher Synthesevergleiche
- Ansatz für Registeroptimierungen
- Zusammenfassung



Bereits vorgestellte Ergebnisse

1. Problem: Hoher Stromverbrauch

Lösungen:

- a) Bis zu 3,5 mal besseres Power-Delay-Produkt als statisch CMOS durch neu entwickeltes AC-TSPC
- b) Weitere Verbesserungen durch Single-Rail-Ansatz und redundante Zahlensysteme
- c) Algorithmus zur Verlustleistungsminimierung von TSPC-Schaltungen in RTL-Ebene



Bereits vorgestellte Ergebnisse

1. Problem: Hoher Stromverbrauch

Lösungen:

- a) Bis zu 3,5 mal besseres Power-Delay-Produkt als statisch CMOS durch neu entwickeltes AC-TSPC
- b) Weitere Verbesserungen durch Single-Rail-Ansatz und redundante Zahlensysteme
- c) Algorithmus zur Verlustleistungsminimierung von TSPC-Schaltungen in RTL-Ebene

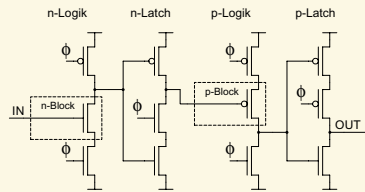
2. Problem: Keine Unterstützung bei der Synthese:

Lösungen:

- DYNAMIC: Entwicklung eines Designflusses für TSPC einschließlich der Integration von Algorithmen zur Verlustleistungsreduktion
- AC-DYNAMIC: Tool zum Aufbau von AC-TSPC und zur Analyse u. Verifikation des Zeitverhaltens

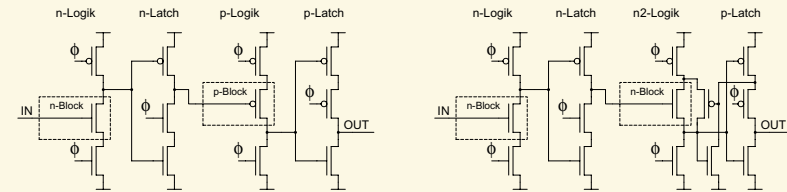


Rückblick: TSPC



nichtinvertierend

Rückblick: TSPC

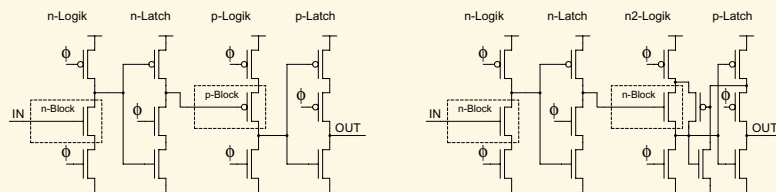


nichtinvertierend

invertierend

Rückblick: TSPC

- Nicht-Inv. Funkt.: Verbrauch bei OUT=0 in der P-, bei OUT=1 in der N-Logik
- Invertierende Funkt.: Verbrauch bei OUT=0 in der N- und in der N2-Logik
- Jedes Gatter realisiert Registerfunktion → extremstes Pipelining
- Einsatzgebiet: höchste Durchsatzraten

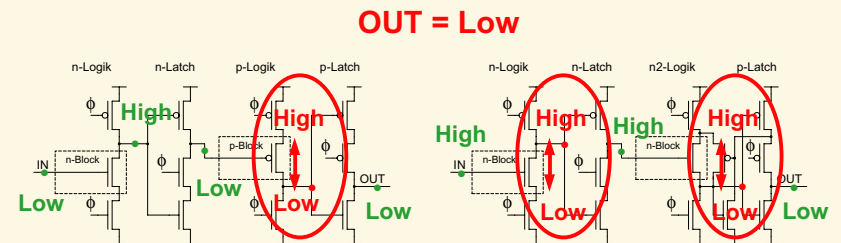


nichtinvertierend

invertierend

Rückblick: TSPC

- Nicht-Inv. Funkt.: Verbrauch bei OUT=0 in der P-, bei OUT=1 in der N-Logik
- Invertierende Funkt.: Verbrauch bei OUT=0 in der N- und in der N2-Logik
- Jedes Gatter realisiert Registerfunktion → extremstes Pipelining
- Einsatzgebiet: höchste Durchsatzraten

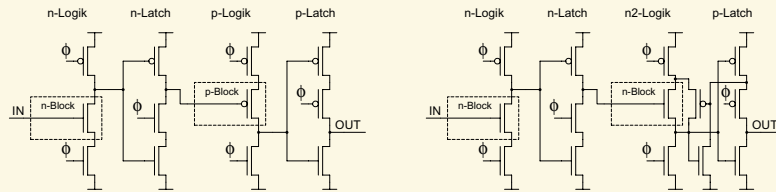


nichtinvertierend

invertierend

Rückblick: TSPC

- Nicht-Inv. Funkt.: Verbrauch bei OUT=0 in der P-, bei OUT=1 in der N-Logik
- Invertierende Funkt.: Verbrauch bei OUT=0 in der N- und in der N2-Logik
- Jedes Gatter realisiert Registerfunktion → extremstes Pipelining
- Einsatzgebiet: höchste Durchsatzraten

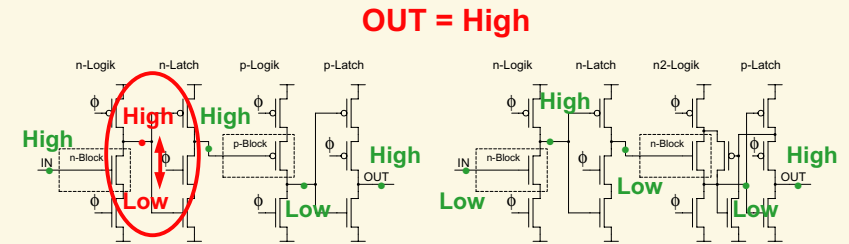


nichtinvertierend

invertierend

Rückblick: TSPC

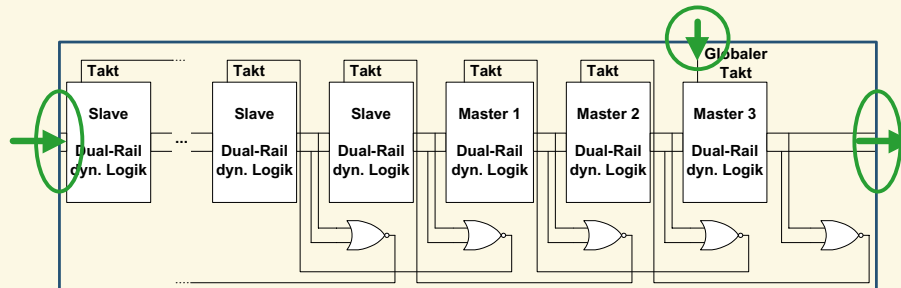
- Nicht-Inv. Funkt.: Verbrauch bei OUT=0 in der P-, bei OUT=1 in der N-Logik
- Invertierende Funkt.: Verbrauch bei OUT=0 in der N- und in der N2-Logik
- Jedes Gatter realisiert Registerfunktion → extremstes Pipelining
- Einsatzgebiet: höchste Durchsatzraten



nichtinvertierend

invertierend

Rückblick: AC – TSPC



- Nach außen synchroner Block
 - Eingangs- und Ausgangssignale
 - Globales Taktsignal
- Kette realisiert Registerfunktion: keine zusätzlichen Latches
- Sehr gutes Power-Delay-Produkt – niedrige Energie
- Einsatzgebiet: niedrige Latenz bei hohen Durchsatzraten

Vergleich (handoptimiert)

Grober Vergleich von verschiedenen Schaltungstechniken durch relative Werte aus verschiedenen Veröffentlichungen:

	Stat. CMOS	Domino	CD-Domino (Sechen)	AC-TSPC (m. Reg.)
Delay (D)	1	0,6	0,4	0,3*
Area	1	1,6	1,5	n.a.*
Power (P)	1	2,6	2,3	1,5*
P*D	1	1,6	0,9	0,5*
Energie*D	1	1	0,4	0,2*

* Design mit Registern

Vergleich (mit Tool, nicht optimiert)

	Stat. CMOS	TSPC	DOMINO	AC-TSPC
Latenz bei ~500MHz (ns)	23,4	31,2	11,7	7,8
Power ohne gl. Takt (mW/GHz)	277	600	686	1361
Energie (10 ⁻⁹ Ws)	3,322	9,610	4,116	5,460
Energie*Delay (10 ⁻¹⁸ J*s)	78	300	48	42
Clockload (minimal Trans.)	~576	~2205	~2034	~370

AC-TSPC, TSPC, DOMINO: 50% zusätzliche Buffer (autom. Synthese, keine Optimierungen); kleine Bibliothek -> reduzierte Eingangslast noch nicht stark bemerkbar

AC-TSPC: nicht optimierte Self-timed Verknüpfungen

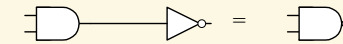
Ergebnisse von 4-Bit Multiplizierern; approximiert für den Vergleichsfall einer Periode von 1,95ns
SCMOS: Durchschnittliche Aktivität: 20%; 96 Register; DOMINO: 48 Register

TSPC: Neuer Ansatz in RTL-Ebene

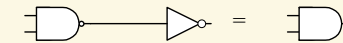
Verwendung angepasster TSPC Gatter:

- Ziel: Ausnutzung, dass ein TSPC N-N2 Gatter keinen Verbrauch bei OUT=1 hat
- langsamster N-Block sollte langsamer sein als P- bzw. N2-Block + stat. Inverter
- Einsparungen stark abhängig von Netzliste

Folgende Ersetzungen erfolgen nach Überprüfung der Aktivität:



TSPC N-P Gatter + Inverter ersetzt ein TSPC N-N2 Gatter

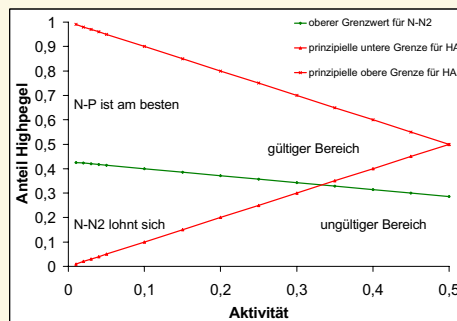


TSPC N-N2 Gatter + Inverter ersetzt ein TSPC N-P Gatter

TSPC: Neuer Ansatz in RTL-Ebene

Verwendung angepasster TSPC Gatter:

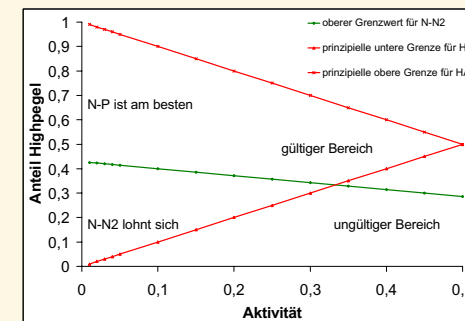
- z.B. nicht-invertierende Logik, wann lohnt sich ein N-N2 Gatter mit nachfolgendem Inverter statt eines N-P Gatters?



TSPC: Neuer Ansatz in RTL-Ebene

Verwendung angepasster TSPC Gatter:

- z.B. nicht-invertierende Logik, wann lohnt sich ein N-N2 Gatter mit nachfolgendem Inverter statt eines N-P Gatters?



Erstes Ergebnis: Reduzierung der Verlustleistung um ca. 10%

Aktuell in Arbeit

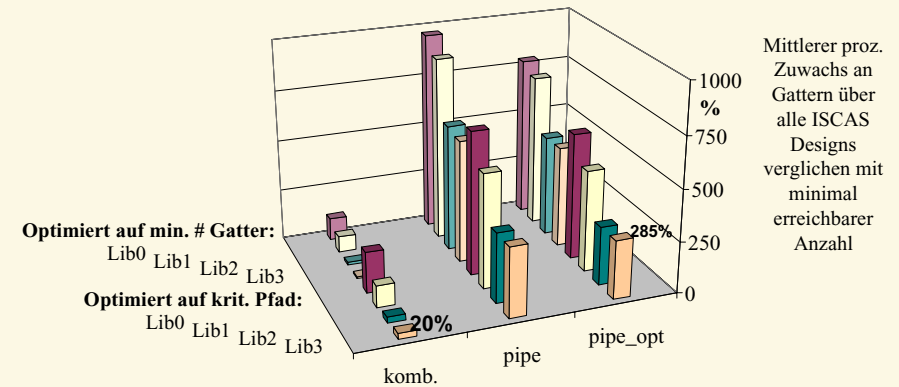
Eignung einer Schaltungstechnik nur in Zusammenhang mit Synthese und Bibliotheksumfang nachweisbar.

Frage: Wie kann ich die Gatteranzahl bzw. die Verlustleistung in der Pipeline reduzieren?

- Synthese: komb. Netzliste → Pipelining (DYNAMIC) → (AC-)TSPC-Netzliste
- Synthese rein kombinatorischer Schaltungen (u.a. ISCAS 85):
 - Variation des Umfangs der Gatterbibliothek
 - Lib0: 4 Gatter (Grundgatter AND, OR mit 2 Eingängen)
 - Lib1: ~ 20 Gatter (Grundgatter mit 2 bis 8 Eingängen)
 - Lib2: ~ 14 Gatter (Grundgatter mit 2 und Komplexgatter mit 2-4 Eingängen)
 - Lib3: ~ 28 Gatter (alle Gatter der anderen Bibliotheken)
 - Syntheseoptionen: minimaler kritischer Pfad bzw. minimale Gatteranzahl
 - DYNAMIC-Optimierungen
- Auswertung von ca. 500 Netzlisten

Vergleich: komb. Netzliste vs. Pipeline

Beste Weg für geringe Gatteranzahl in AC-TSPC bzw. TSPC:
umfangreiche Bibliothek & Optimierung auf kurzen kritischen Pfad



Anteil Register

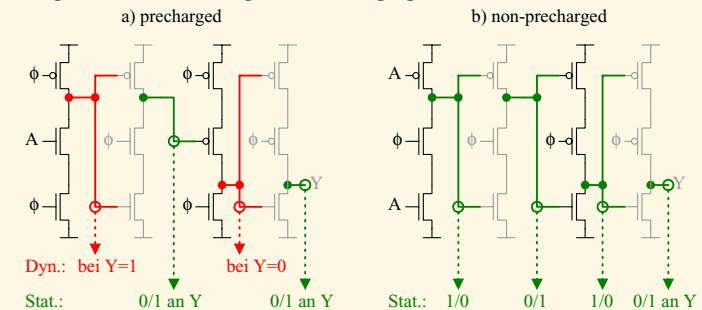
Anteil der Register ohne logische Funktion an Gesamtgatteranzahl: \emptyset 65%!

- Ziel: Reduzierung der Registeranzahl, Optimierung der Schaltung des Registers

Synthese	Anteil Register (%)		
	min.	\emptyset	max.
Optimiert	52 (c6288)	65	76 (c5315)

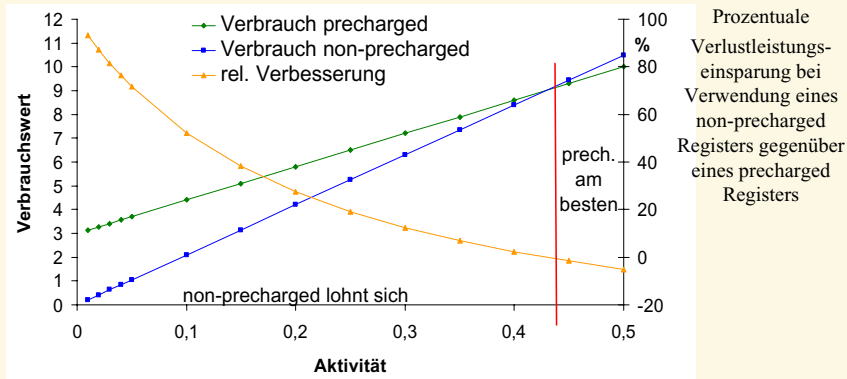
Neu: Optimierung Register/Buffer

- AC-TSPC: Ersetzen von Ketten ohne Verzweigungen, die nur Buffer enthalten, mit einem TSPC-Register
- TSPC: Nutzung eines non-precharged Registers:
 - Voraussetzung: mind. gleiche Geschwindigkeit von non-precharged Register und dem langsamsten Logikgatter



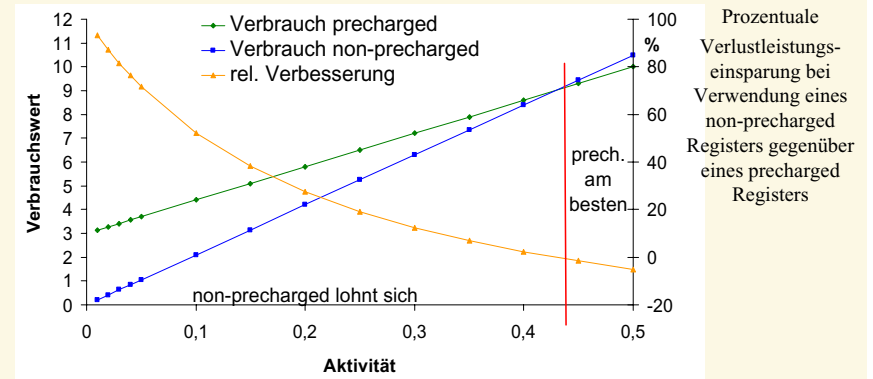
Neu: Optimierung TSPC-Register

Wann ist ein precharged Register günstiger?



Neu: Optimierung TSPC-Register

Wann ist ein precharged Register günstiger?



Erstes Ergebnis inkl. Gatteraustausch: Verlustleistungsreduzierung auf ca. 85%

Ausblick

- Fortsetzung der Untersuchungen mit umfangreicher Anzahl an Testdesigns:
 - Simulation der Aktivität → Potential der Verbesserungen auf RTL-Ebene
 - Testung der Verlustleistungsabschätzung
 - Vervollständigung der Syntheseunterstützung für Domino/DCVSL
- Vergleich von verschiedenen Schaltungstechniken: Domino, TSPC, AC-TSPC und statisch CMOS
- ➔ Festlegung besonders günstiger Designeigenschaften für dynamische Schaltungen
- Simulation auf Transistorebene für einige Beispielschaltungen
- Untersuchungen für die Gültigkeit der Ergebnisse in zukünftigen sub-micron Technologien

Zusammenfassung

- Ausarbeitung und Erweiterung der AC-TSPC Logik durch Single-Rail Ansatz mit redundanten Zahlensystemen
 - F. Grassert, D. Timmermann, „Dynamic Single-rail Self-timed Logic Structures for Power Efficient Synchronous Pipelined Designs“, GLSVLSI 2003
 - F. Grassert, D. Timmermann, „Dynamic Self-timed Logic Structures“, DDECS 2003
- Aufbau von Schaltungen
 - Zusammenhänge zwischen kombinatorischer Netzliste und daraus erstellter Pipeline
 - Einfluss der Gatterbibliothek
- Verlustleistungsreduzierung durch Veränderungen an Register und Austausch von TSPC Gattern
- Längerfristige Kooperation mit Halbleiterherstellern bzgl. Low Power und dynamischen Schaltungstechniken

Veröffentlichungen seit 2001

- [Flü01] S. Flügel, F. Grassert, M. Grothmann, M. Haase, P. Nimsch, H. Ploog, D. Timmermann, A. Wassatsch, „A Design Flow for 1.28 Gbit/s Triple DES using dynamic logic and standard synthesis tools“, Synopsys User Group (SNUG) Europe, S. E3.2. 1-8, München, März 2001.
- [Was01] A. Wassatsch, F. Grassert, M. Grothmann, D. Timmermann, „Analyse von Synthese-Algorithmen für Pipeline-Strukturen“, Fachtagung Informationstechnik, Universität Magdeburg, ISBN: 3-930385-29-5, S. 17-22, Magdeburg, März 2001.
- [Gra01a] F. Grassert, D. Timmermann, „Dynamic Single Phase Logic with Self-Timed Stages for Power Reduction in Pipeline Circuit Designs“, IEEE International Symposium on Circuits and Systems (ISCAS), Sydney, Mai 2001.
- [Gra01b] F. Grassert, D. Timmermann, „Integration asynchroner Schaltungsansätze zur Verringerung der Verlustleistung in dynamischen Schaltungstechniken mit Einphasentakt“, 10. Symposium Maritime Elektronik, S. 197-200, Rostock, Juni 2001.
- [Gra01c] F. Grassert, A. Wassatsch, D. Timmermann, „Synthese- und Einsatzmöglichkeiten asynchroner, dynamischer Logiken zur Verlustleistungsreduzierung“, 14. Mikroelektroniktagung 2001, ÖVE-Schriftenreihe Nr.26, S.231-236, Wien, Oktober 2001.
- [Gra02a] F. Grassert, D. Timmermann, „Asynchronous Chain True Single Phase Clock Logik (AC-TSPC)“, 3. Schwerpunktkolloquium des DFG Schwerpunktprogramms Grundlagen und Verfahren verlustarmer Informationsverarbeitung (VIVA), S. 136 - 141, Chemnitz, März 2002.
- [Gra02b] F. Grassert, D. Timmermann, „Single-Rail Self-Timed Logic Circuits in Synchronous Designs“, IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'2002), Tulsa, August 2002.
- [Bra03] R. Brackebusch, S. Müller, G. Sokomak, F. Grassert, D. Timmermann, „A New Synthesizable Architecture Approach for Verification Environments Applying Transaction-based Methodology“, E.I.S. 2003 - Entwurf Integrierter Schaltungen und Systeme, Erlangen, März/April 2003
- [Gra03a] F. Grassert, D. Timmermann, „Dynamic Single-rail Self-Timed Logic Structures for Power Efficient Synchronous Pipeline Designs“, Great Lakes Symposium on VLSI, Washington, April 2003
- [Gra03b] F. Grassert, D. Timmermann, „Dynamic Self-timed Logic Structures“, DDECS 2003
- [Sii03] F. Sill, F. Grassert, A. Wassatsch, D. Timmermann, „A Design Flow for Asynchronous Dynamic Logic and Standard Synthesis Tools“, SNUG, Boston, September 2003

