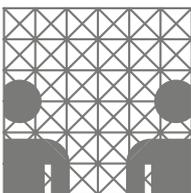


Praktikum: VLSI-Entwurf

2

Full-Custom Design dynamische Register



Andreas Mäder

Universität Hamburg – MIN – Fachbereich Informatik
Arbeitsbereich Technische Aspekte Multimodaler Systeme

<http://tams.informatik.uni-hamburg.de>

In diesem Versuch sollen Sie lernen, wie dynamische Schaltungen funktionieren. Außerdem wird eine Schaltungstechnik vorgestellt, die bei Einsatz eines sehr einfachen Taktschemas den Aufbau von sehr schnellen logischen und speichernden Elementen ermöglicht.

Im Rahmen der Entwurfssoftware werden dabei Werkzeuge zur Layoutsynthese eingesetzt, die es ermöglichen, Schaltungen in Full-Custom Qualität aus Transistornetzlisten zu generieren.

Voraussetzungen

- Grundwissen zu dynamischen Schaltungen — wäre wünschenswert, ansonsten folgt gleich noch eine kurze Einführung.

Dynamische und statische Schaltungen

Bei den Versuchen hier arbeiten Sie mit *dynamischen Schaltungen*, das heißt, dass die Gates nachfolgender Transistoren über getaktete Transistoren (die als Schalter wirken) aufgeladen bzw. entladen werden. Anschließend wird durch Sperren dieser Transistoren erreicht, dass die auf diesen Gates aufgebrachte Ladung eine gewisse Zeit gespeichert bleibt. Wegen der Selbstentladung muss die Schaltung allerdings nach einiger Zeit wieder getaktet werden.

Je nach Schaltungstechnik lassen sich dabei zwei Funktionsweisen unterscheiden:

1. die aufgeladenen Gates werden elektrisch isoliert und wirken als „reiner“ Kondensator.
2. abhängig von den Eingängen der Schaltung, werden die aufgeladenen Gates entweder entladen, oder die gespeicherte Ladung bleibt erhalten. Über die Eingänge wird dabei ein logischer Ausdruck, wie bei statischen CMOS-Schaltungen, gebildet.

Das Aufbringen der Ladung auf die Gates und das anschließende Isolieren dieses „Kondensators“ entspricht einer dynamischen Speicherung von Information.

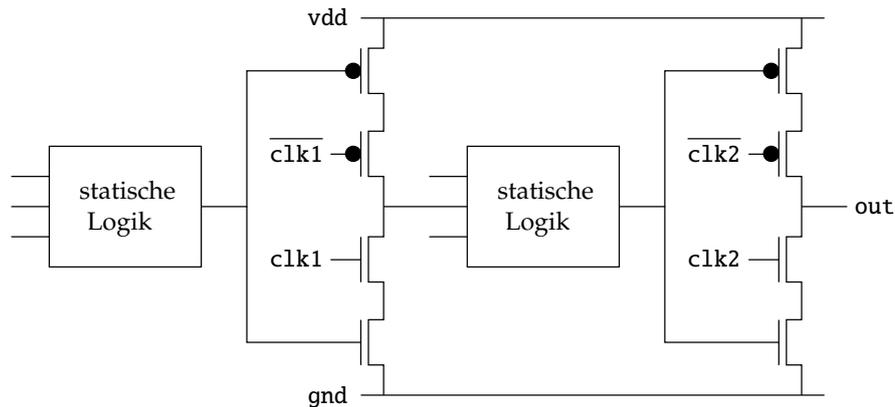
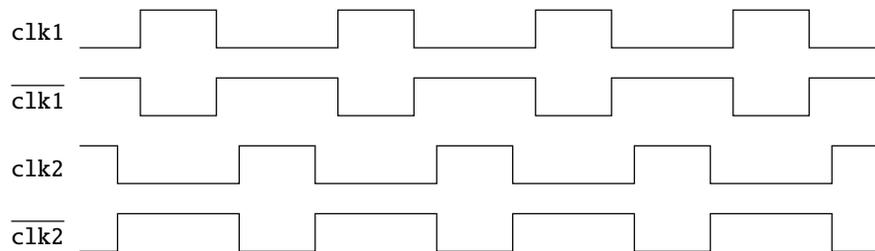
Im Gegensatz dazu wird bei *statischer Technik* eine Speicherzelle (wie zum Beispiel ein Basis-RS Flipflop aus zwei NOR-Gattern) verwendet, die die gespeicherte Information beliebig lange hält. Aufbauend auf solchen Speicherzellen werden dann, entsprechend dem Taktverhalten, die Flipflops (taktpegel-, vorderflanken-, rückflanken-, zweiflankengesteuert, master-slave ...) entworfen.

Dynamische Schaltungstechniken und Pipelinerealisierung

Der Vorteil der dynamischen Technik liegt darin, dass die Speicherung von Information quasi nebenbei abfällt. Die Unterscheidung in die zwei Betriebsarten: *Aufladephase* und *Speicher-/Auswertungsphase* kostet, im Vergleich zu statischen Schaltnetzen, nur scheinbar Zeit, da mit dynamischen Schaltungen extrem hohe Taktraten möglich sind.

Mehrstufige arithmetische Pipelines (Logik → Register → Logik → Register...) lassen sich in dynamischer Technik sehr einfach realisieren. Neuere Schaltungstechniken ermöglichen es, die beiden Betriebsphasen auf wahlweise beide Taktpegel zu legen, es gibt dann jeweils zwei Schaltungsvarianten:

	Aufladephase	Auswertungsphase
1.	Clock = 0	Clock = 1
2.	Clock = 1	Clock = 0

Abbildung 1: C²MOSAbbildung 2: Taktschema für C²MOS

Ordnet man in einer arithmetischen Pipeline die beiden Schaltungsvarianten in aufeinander folgende Stufen abwechselnd an, so wird die „gesamte“ zur Verfügung stehende Zeit eines Taktsignals ausgenutzt.

Anschließend werden einige Schaltungsvarianten vorgestellt. Zum Verständnis der Funktionsweise sollte man sich jeweils vergegenwärtigen, wann einzelne Transistoren leiten, bzw. sperren.

Clocked CMOS: dies ist die „älteste“ Schaltungstechnik.

- | | |
|----------------|---|
| Funktionsweise | – Die Gates von statisch aufgebauter Logik werden für die Speicherung benutzt. |
| | – Durch die zwei disjunkten Takte wird ein „geordnetes“ Durchschieben der Werte erreicht. |
| Nachteile | – 2 Takte: komplizierter Clock-Generator |
| | – 4 Taktleitungen: Probleme beim Layout |
| | – Totzeiten: niedrige Taktraten |
| Probleme | – das Taktschema muss genau eingehalten werden (Clock-Delay) |

NORA (NO RAcE 2-Phasen Takt):

- | | |
|----------------|--|
| Funktionsweise | – Das Speicherelement (Latch) von C ² MOS wird zusammen mit dynamischer Logik eingesetzt. |
| | – Für den Aufbau von Pipelines stehen zwei Schaltungsvarianten, P- und N-Stufen, zur Verfügung. |

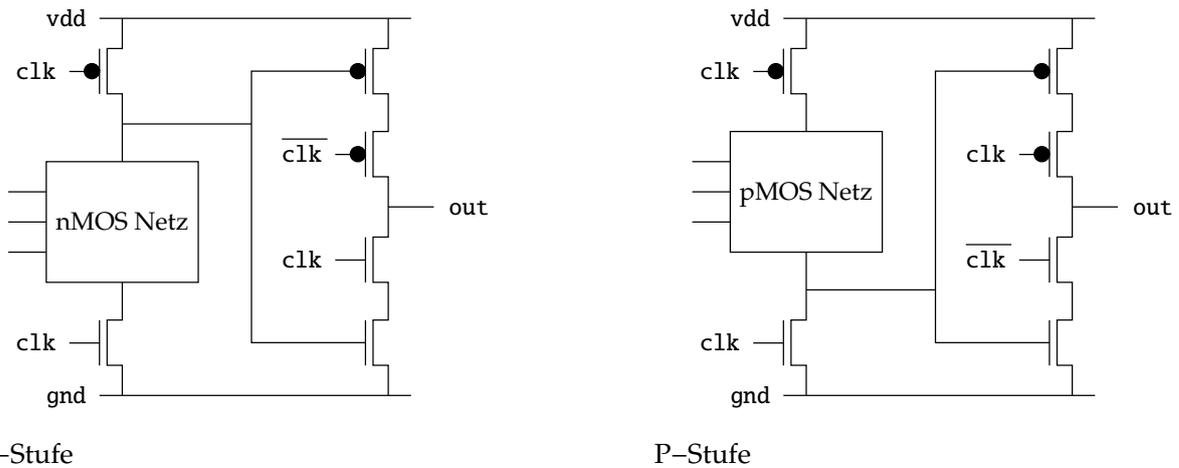


Abbildung 3: NORA

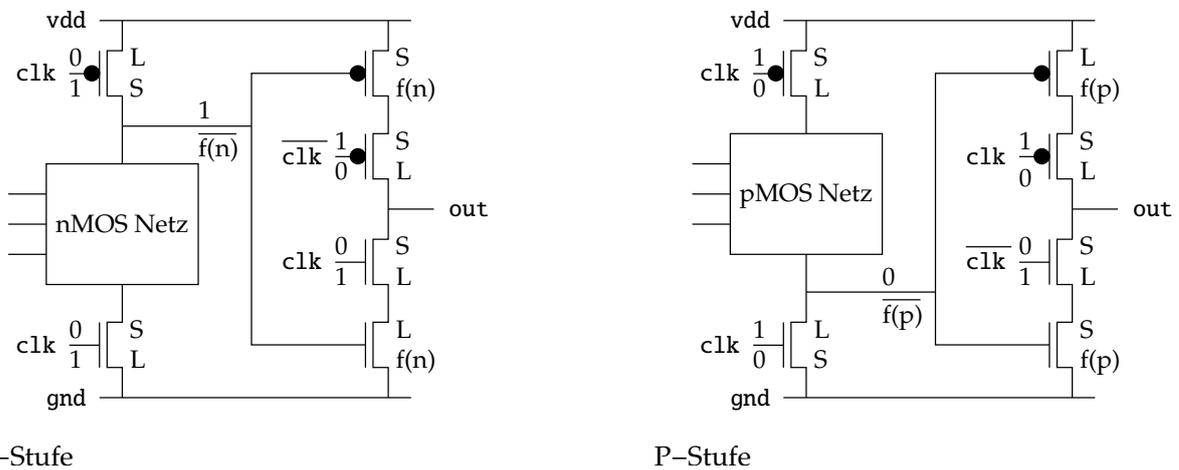


Abbildung 4: Funktionsweise von NORA

- | | |
|-----------|--|
| Vorteile | – durch dynamische Technik werden bei größeren Gatterfunktionen Transistoren eingespart (weniger Fläche) |
| | – da keine Totzeiten auftreten werden höhere Taktraten erreicht |
| Nachteile | – 2 Taktleitungen: Probleme beim Layout |
| Probleme | – Vorder- und Rückflanke des Takts müssen „gleichzeitig“ wechseln |

TSPC (True Single Phase Clock): diese Schaltungstechnik hat sich ab Anfang der 90er bei den ersten „schnellen Prozessoren“ (z.B.: DEC Alpha, PowerPC) durchgesetzt.

- Funktionsweise – Betrachtet man die Funktionsweise von NORA (Abb. 4) so stellt man fest, dass die Transistoren, die mit dem invertierten Takt betrieben werden, wegfallen können.
- Für den Aufbau von Pipelines stehen P- und N-Stufen zur Verfügung, die abwechselnd, bzw. nach Kombinationsregeln, angeordnet werden.

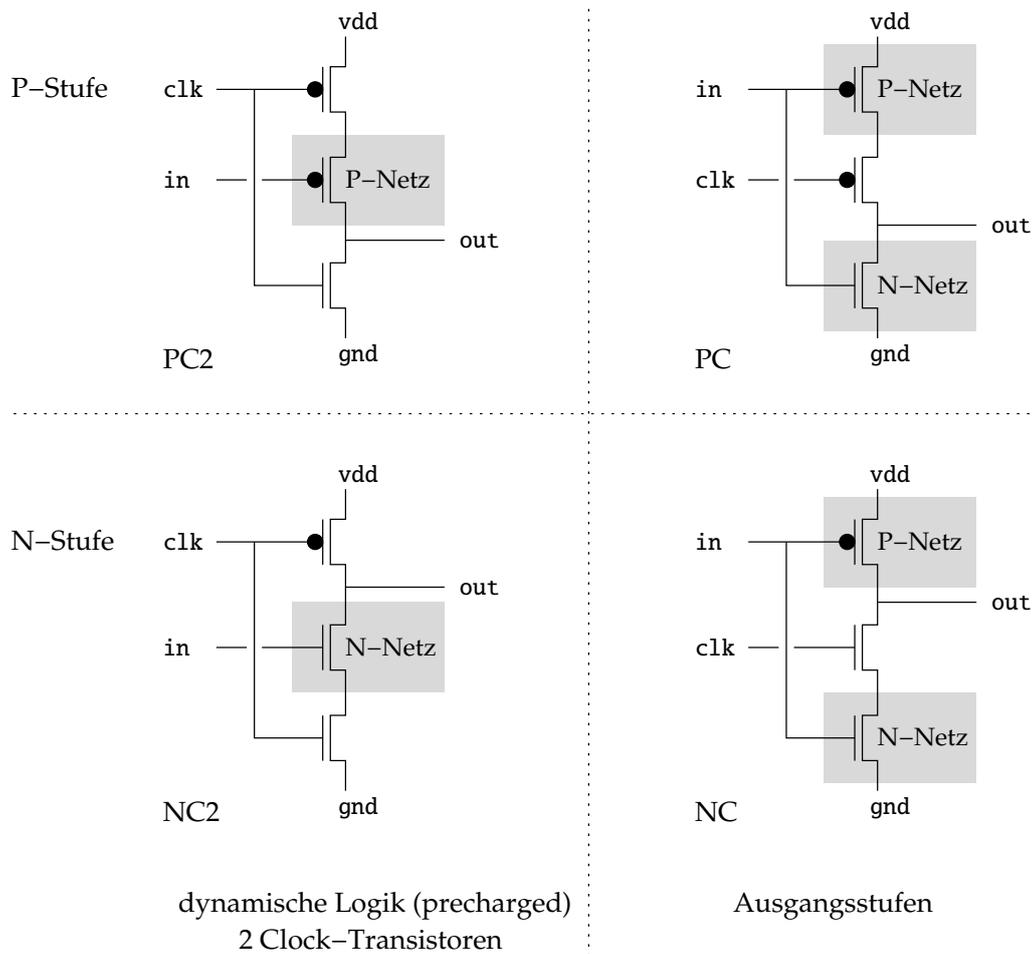
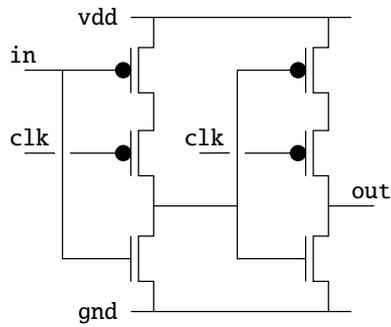


Abbildung 5: TSPC-Grundelemente

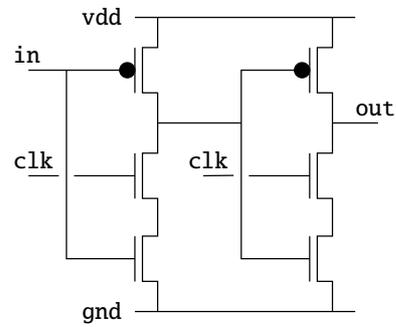
- | | |
|----------|--|
| Vorteile | <ul style="list-style-type: none"> – da keine Totzeiten auftreten werden sehr hohe Taktraten erreicht (> 400 MHz bei einem „alten“ 0,7 μm Prozess, mehrere GHz bei aktuellen Submikron-Prozessen) – es gibt nur eine Clock (-Leitung) |
| Probleme | <ul style="list-style-type: none"> – die Taktsignale müssen sehr steile Flanken besitzen – der Clock muss bei allen getakteten Transistoren „gleichzeitig“ eintreffen – die Kombinationsregeln zwischen aufeinander folgenden P- und N-Stufen sind <i>genau</i> einzuhalten |

Für den Aufbau von TSPC Schaltungen stehen die vier Grundelemente aus Abb. 5 zur Verfügung. Dabei lassen sich P- und N-Netze durch die Realisierung logischer Ausdrücke ersetzen — die hier eingezeichneten Transistoren entsprechen einfachen Inverterfunktionen.

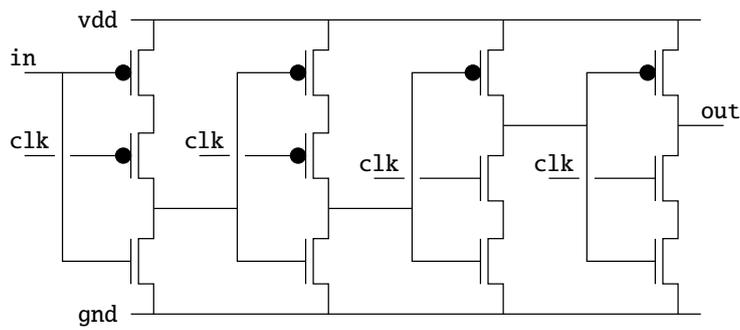
Durch die Kombination dieser Grundelemente lassen sich so verschiedene Latches und Flipflops aufbauen (Abb. 6 und Abb. 7) — anhand der Schaltzustände der Transistoren kann man sich die Funktion veranschaulichen.



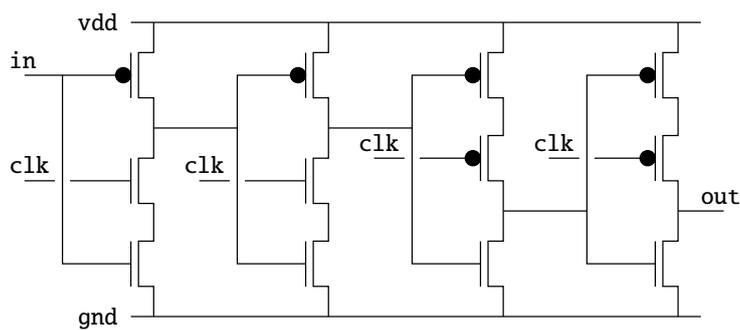
Latch – low transparent PC-PC



Latch – high transparent NC-NC

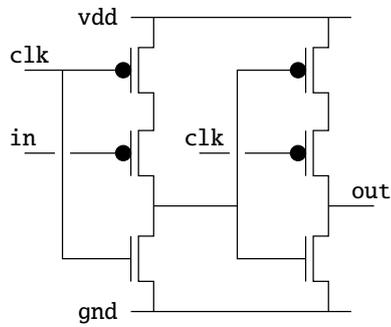


Flipflop – vorderflankengesteuert PC-PC-NC-NC

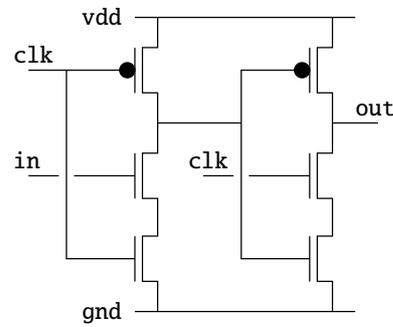


Flipflop – rückflankengesteuert NC-NC-PC-PC

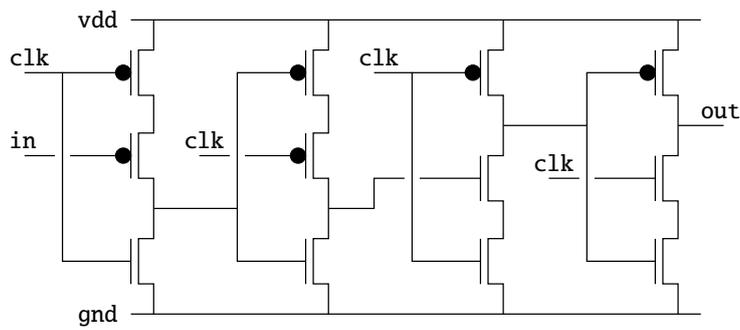
Abbildung 6: TSPC-Latches und -Flipflops — non-precharged



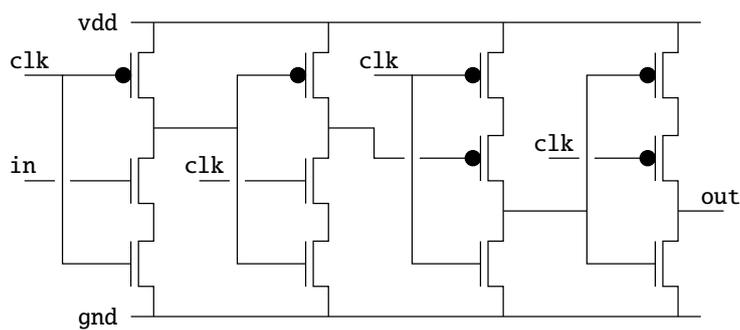
Latch – low transparent PC2-PC



Latch – high transparent NC2-NC



Flipflop – vorderflankengesteuert (PC2)-PC-NC2-NC



Flipflop – rückflankengesteuert (NC2)-NC-PC2-PC

Abbildung 7: TSPC-Latches und -Flipflops — precharged

Arbeitsweise

Die Schritte entsprechen in ihrer Reihenfolge dem Entwurfsvorgehen, wie es in „Layoutsynthese“, bzw. „Full-Custom Layout“, beschrieben ist.

1. Entwerfen Sie die Transistornetzliste der Schaltung mit dem Schematic-Editor und generieren Sie ein Symbol für den Anschluss an die Testumgebung.
2. Zur Prüfung der Schaltung müssen Sie jetzt eine Testumgebung entwerfen.
3. Anschließend wird das Design simuliert.
4. (optional) Erzeugen Sie ein Layout der Netzliste mit Hilfe der Layoutsynthese- und Kompaktierungswerkzeuge.

Aufgaben

II-1 Entwerfen Sie ein **Latch (high transparent)** in TSPC-Technik und überprüfen Sie die Korrektheit der entworfenen Schaltung durch geeignete Simulation.

Fragen:

- Welche Unterschiede bestehen zwischen den „non-precharged“ und den „precharged“ Realisierungen?

Die Schaltbilder sind in den Abbildungen 6 und 7 dargestellt.

II-2 Entwerfen Sie ein **vorderflankengesteuertes D-Flipflop** in TSPC-Technik und überprüfen Sie die entworfene Schaltung durch geeignete Simulation.

Fragen:

- Wie schnell kann das Flipflop getaktet werden?
- Welche Zeitbedingungen sind einzuhalten, damit der Entwurf noch funktioniert?

Die Schaltbilder sind in den Abbildungen 6 und 7 dargestellt.