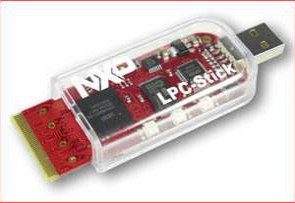


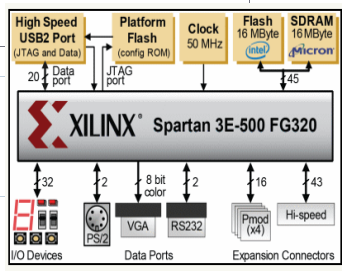
Informatik  
HAW Hamburg



Hochschule für Angewandte Wissenschaften Hamburg  
Hamburg University of Applied Sciences

## Computer Engineering WS 2012

Digitale Systeme  
Pipeline



Prof. Dr. B. Schwarz

Informatik  
HAW Hamburg

## Digitale Systeme

Hochschule für Angewandte Wissenschaften Hamburg  
Hamburg University of Applied Sciences

### 3. Architektursynthese

#### ASM-Diagramme

**Prozessorelement für eine S-Kurvenapproximation**

**Gemeinsame Nutzung von Hardware Funktionseinheiten**  
(Ressource Sharing)

**Gemeinsame Register- / Speicher-Nutzung**  
(Register Sharing)

**VHDL-Modell des PE mit Fixed-Point Arithmetik im Q-Format**

#### Datenpfad in Pipelinestruktur

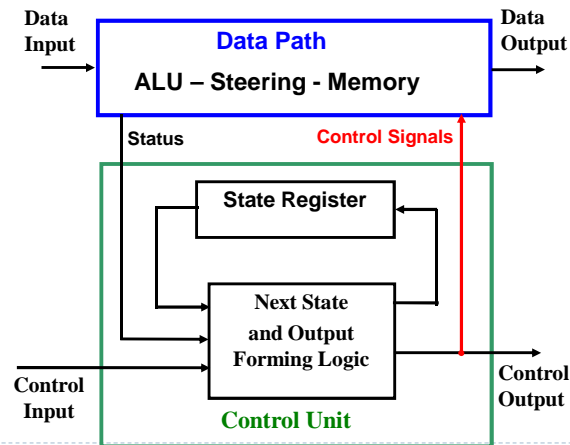
**Dynamische Systeme mit Rückführung**

**Multizyklusdatenpfad für ein Filter 1. Ordnung**

**Ergebnisübergabe an Folgezyklen**



## Strukturierung eines digitalen Systems als Prozessorelement in einen Datenpfad und einen Steuerepfad



3

CE - DS 3.3 Pipeline



## Pipelining

- Pipelining ist das **Aufbrechen langer Signalpfade** in etwa gleich lange Schaltnetzstufen.
- Eine Separierung der **N Logik-Teilabschnitte** im Daten- (und ggf. im Steuerepfad) erfolgt mit **Pipeline-Registern**.
- Jeder einzelne Schaltnetzabschnitt wird taktsynchron gespeichert. Dadurch werden **N unterschiedliche Eingangsdaten (ein Datensatz) parallel** in aufeinander folgenden Stufen verarbeitet
- Wenn die Länge der verkürzten N Signalpfadabschnitte etwa gleich groß ist und die **Laufzeit** jedes einzelnen  $T_{CLK}/N$  der ursprünglichen Taktperiode  $T_{CLK}$  entspricht, so ist die Bearbeitungsdauer (Latenz) für ein Eingangsdatum nahezu genauso lang wie ohne Pipelining.
- Allerdings ist der **Datendurchsatz mit Pipelining größer**, da pro Intervall  $T_{CLK}/N$  ein aktualisierter Ausgang verfügbar wird!
- Bis ein Ergebnis am Ausgang liegt vergeht eine **Latenz von N-Takten ( $N \cdot T_{CLK}/N$ )**.

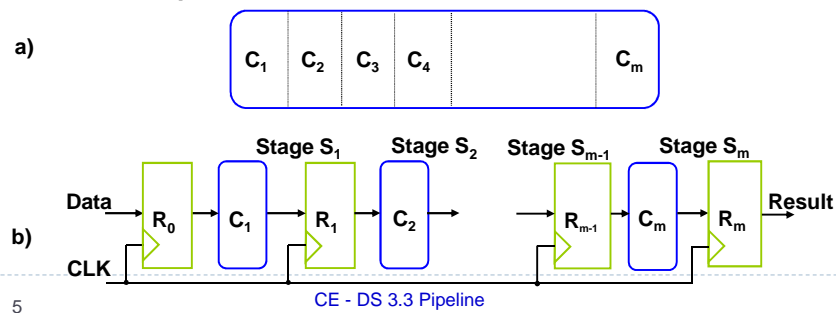
4

CE - DS 3.3 Pipeline



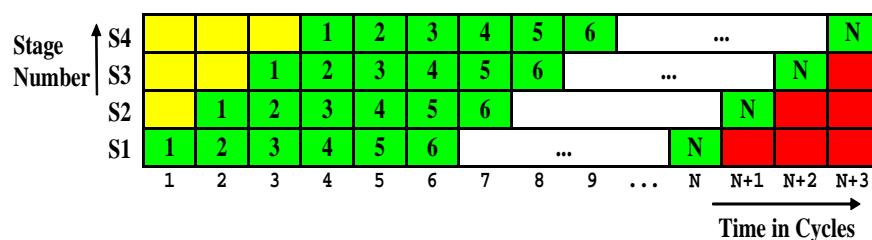
## Pipelining of Data Paths

- Pipelining increases performance by restructuring long data paths with several **levels of logic ( $C_i$ )** and breaking it up over multiple clock cycles. Pipelining adds registers in the combinational logic. **Shorter clock cycle supports an increased data throughput at the expense of added data latency.**
- A computational circuit has to be partitioned into several approximately equal delay parts ( $C_i$ ) and then inserting registers in between the partitions.



## Phasendiagramm (1)

- For visualizing the operation of a pipeline a **space-time diagram** is used. The horizontal axis corresponds to time and the vertical axis corresponds to the stage number. The entries in the green boxes correspond to the symbolic IDs of the data currently worked on.
- The shown space-time diagram describes an example of a  **$m=4$**  stage pipeline, with **each stage clocked four times** the frequency of the system without pipelining. After  **$N+3$**  rising clock edges  **$N$**  data items are completely processed.



## Phasendiagramm (2)

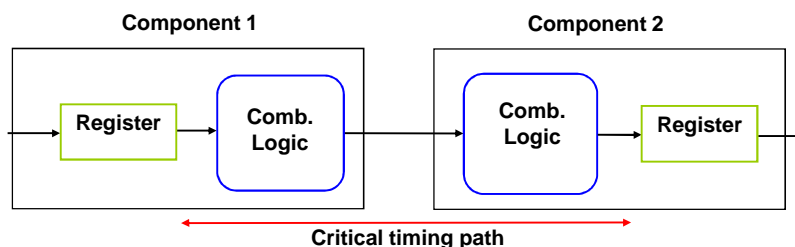
- Looking at one time instant, i.e. time unit 4, we can see that all stages of the pipeline are busy working on different data items. A data item is completely processed when it leaves the last stage. The first data item leaves the pipeline when all stages are filled. After that has accomplished **each single cycle yields a new data item result**.
- Disregarding the first  $m-1=3$  cycles the pipeline works with a speedup which is determined by the number  $m$  of the stages: four times.
- In general, if there are  **$m$  stages** in the pipeline, the time taken to generate results can be **reduced to  $1/m$  times** of the non-pipelined execution time, with the exception of the first result.

7

CE - DS 3.3 Pipeline

## Control Pipelining (1)

- A critical timing path may arise in control and status signal lines between data path and control unit.
- The figure depicts the direct interfacing of an **FSM output forming logic** to an **input decoding logic of a data path**. Coupled combinational circuits of connected components have to be broken up with inserted registers in order to shorten data paths.

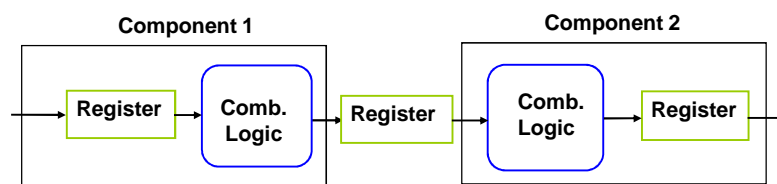


8

CE - DS 3.3 Pipeline

## Control Pipelining (2)

- Each design cycle has to be finished with a post layout timing simulation which will yield an analysis of frequency limiting timing paths.
- After each step of inserting pipelining registers into data path and/or component connections special considerations must be applied to the number of states in control FSM and their transitions. The overall timing has to fit to the added path latency.



9

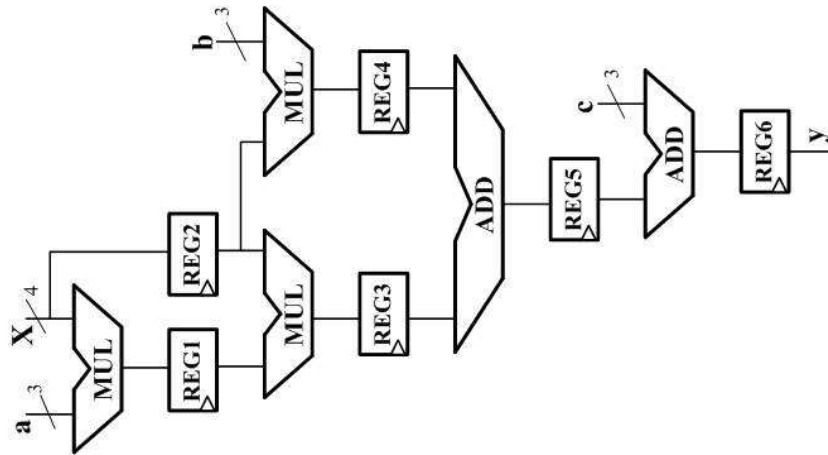
CE - DS 3.3 Pipeline

## Pipelining-Struktur für die S-Kurvenapproximation

10

CE - DS 3.3 Pipeline

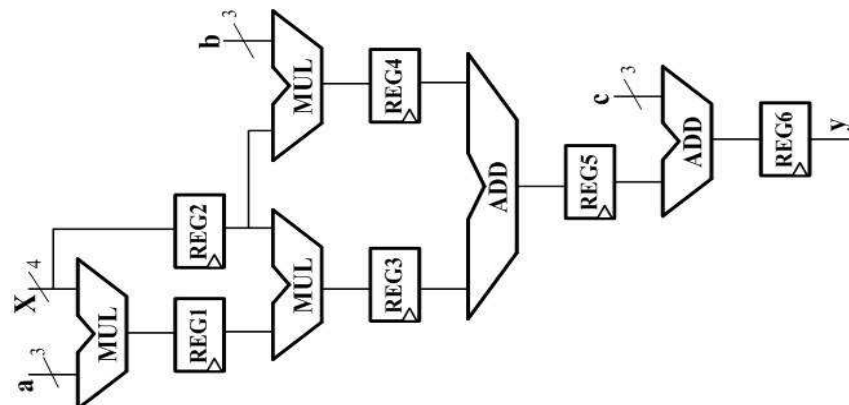
## Polynomauswertung



11

CE - DS 3.3 Pipeline

## Polynomauswertung mit zeitveränderlichen Parametern



12

CE - DS 3.3 Pipeline