

CISC vs RISC

Fließbandverarbeitung

CISC vs. RISC und die Performance von Rechnern

Der Befehlsteilheit

Rechenleistung

Fließbandverarbeitung

CISC vs RISC

Befehlszyklus

V. Neumann - Rückkehr



Rechenfertigkeit

Performance eines Rechners (engl.: to perform = ausführen)

Die Performance (Performance) eines Rechners entspricht seiner Rechenleistung. Aus Sicht eines Anwenders ist dies der Durchsatz an Daten (Datendurchsatz). Da der Datendurchsatz proportional zu Befehlsdurchsatz ist, letzteres aber einfacher zu messen ist, ist die Leistung eines Rechners durch seine Befehlsdurchsatz, also die Zahl der Befehle, die pro Zeitseinheit ausgeführt werden können definiert.

Damit ist der Durchsatz zu einer Abhängig von der Zyklenanzahl

$$T_c = \frac{1}{fc}$$

mit der höchstmöglichen Taktfrequenz.

und der Anzahl der pro Befehl benötigten Taktzyklen.

Gekl. Laufzeit (zur Eliminierung: Aus GdTI bekannte)

Springantwort $U_a(t) = U_e(0) e^{-\frac{t}{\tau}}$ $U_a(t) \approx U_e(0) (1 - e^{-\frac{t}{\tau}})$

Bitwechsel ($L \rightarrow H$ v $H \rightarrow L$) bei $U_a(t) = \frac{1}{2} U_e(0)$

$$\cancel{U_e(0)} e^{-\frac{t}{\tau}} = \frac{1}{2} \cancel{U_e(0)} \quad \cancel{U_e(0)} (1 - e^{-\frac{t}{\tau}}) = \frac{1}{2} \cancel{U_e(0)}$$

$$-\frac{t}{\tau} = \ln\left(\frac{1}{2}\right)$$

$$-t = \tau (-\ln(2))$$

$$\ln(1) - \frac{t}{\tau} = \ln(2^{-1})$$

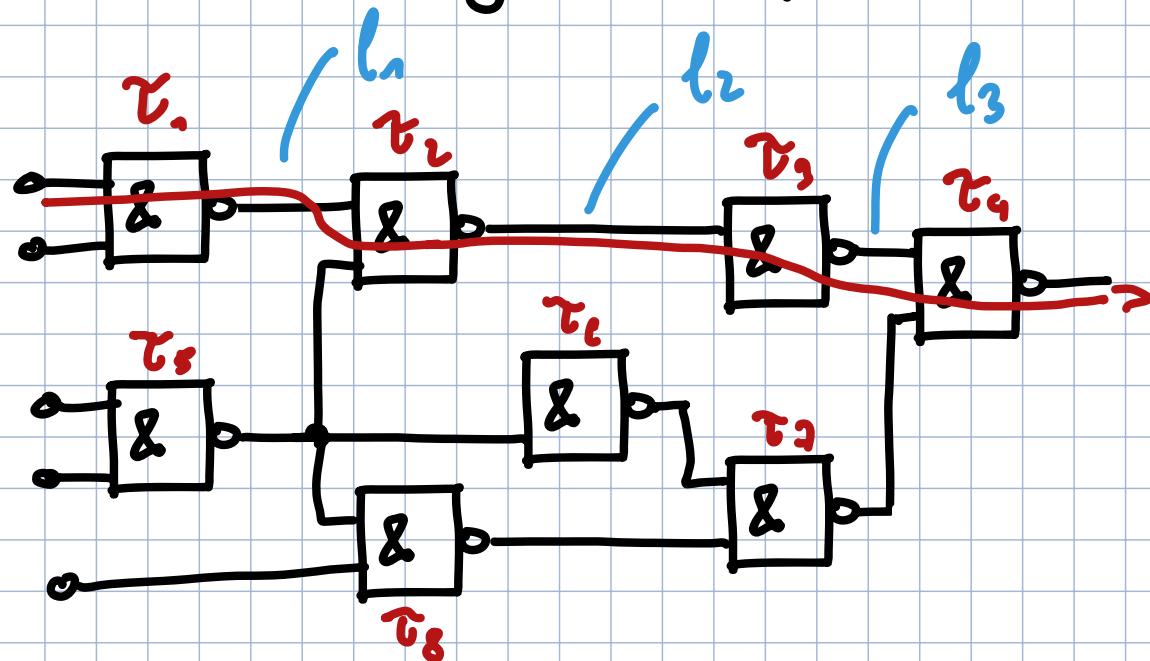
$$\cancel{\ln(1)} + \ln(2) = \frac{t}{\tau}$$

$t = \tau \ln(2)$

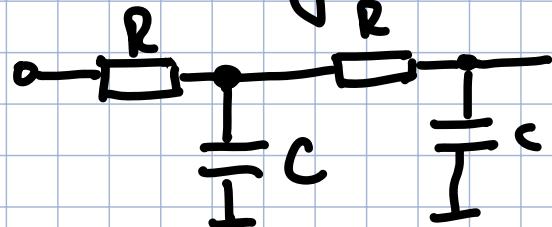
$t = -\tau \ln(2)$

$$\tau = RC$$

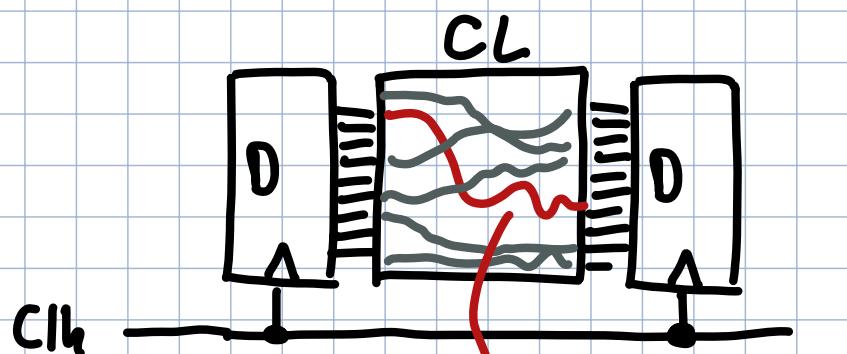
Prozessortakt (technologische Griffe) :



Modell der Leitungen:



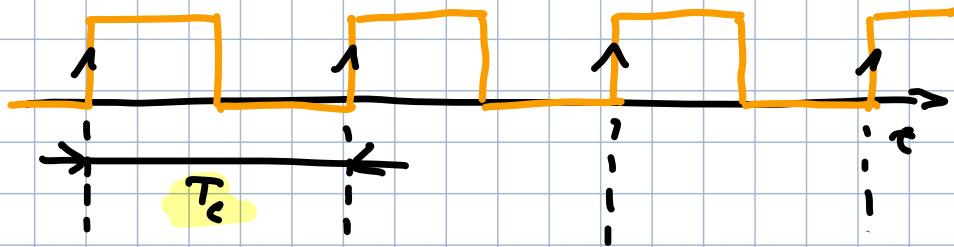
Auf IC's kann die Induktivität vernachlässigt werden.



längster logischer Ausführungspfad

$$\sum_{\tau \in \#} \tau_i + \sum_{i \in L} l_i = T_c$$

Taktzeit



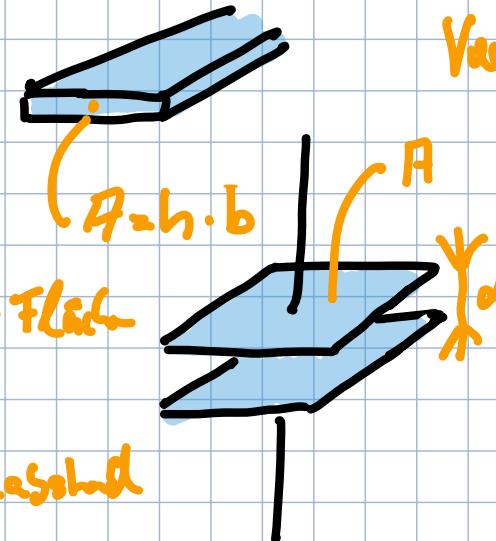
Technologische Shalting:

Elektrischer Widerstand:

$$R = \rho \frac{\ell}{A}$$

Elektrische Kapazität:
(Kondensator)

$$C = \epsilon_0 \epsilon_r \frac{A}{d} - \text{Fläche}$$



Variabler Spaltwiderstand
Plattenkondensator
Spieldauer

$$\tau = RC \ln(2)$$

Shalting (nach Moores Gesetz) verringert
auf automatisch R und C ! Damit wird
die Schaltung von selbst schneller!

| Shalting: | SSI | LSI | VLSI | ULSI | 2010 | 2020 |
|-----------------|------|------|------|------|-------|-------|
| Strukturgruppe: | 1960 | 1970 | 1980 | 1990 | 65 nm | 32 nm |
| Technologie: | PMOS | NMOS | CMOS | | 10 nm | |

Vervesserung der Performance durch Architektur (Iron Law)

$$\frac{1}{P} := n_p \cdot CPI \cdot T_c$$

Diagram illustrating the components of the Iron Law formula:

- Computer**: Points to the term n_p .
- Befehlsatz-Architektur**: Points to the term CPI .
- ISA**: Points to the term T_c .
- Technologie**: Points to the term T_c .

Instruktion per Cycle:

$$I_c = \frac{1}{CPI}$$

Entwurfssiel:

$$CPI < 1$$

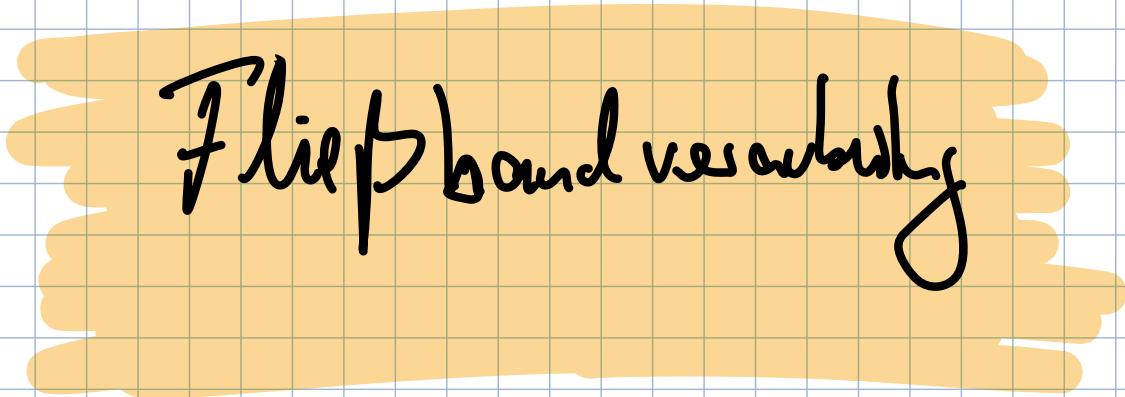
CPI = Cycles per Instruktion
 n = Anzahl der Befehle
 T_c = Befehlsvzykluszeit

Typische Prozessor 1980:

$$\bar{CPI} \approx 10$$

Entwurfssiel bis 1990

$$CPI = 1$$

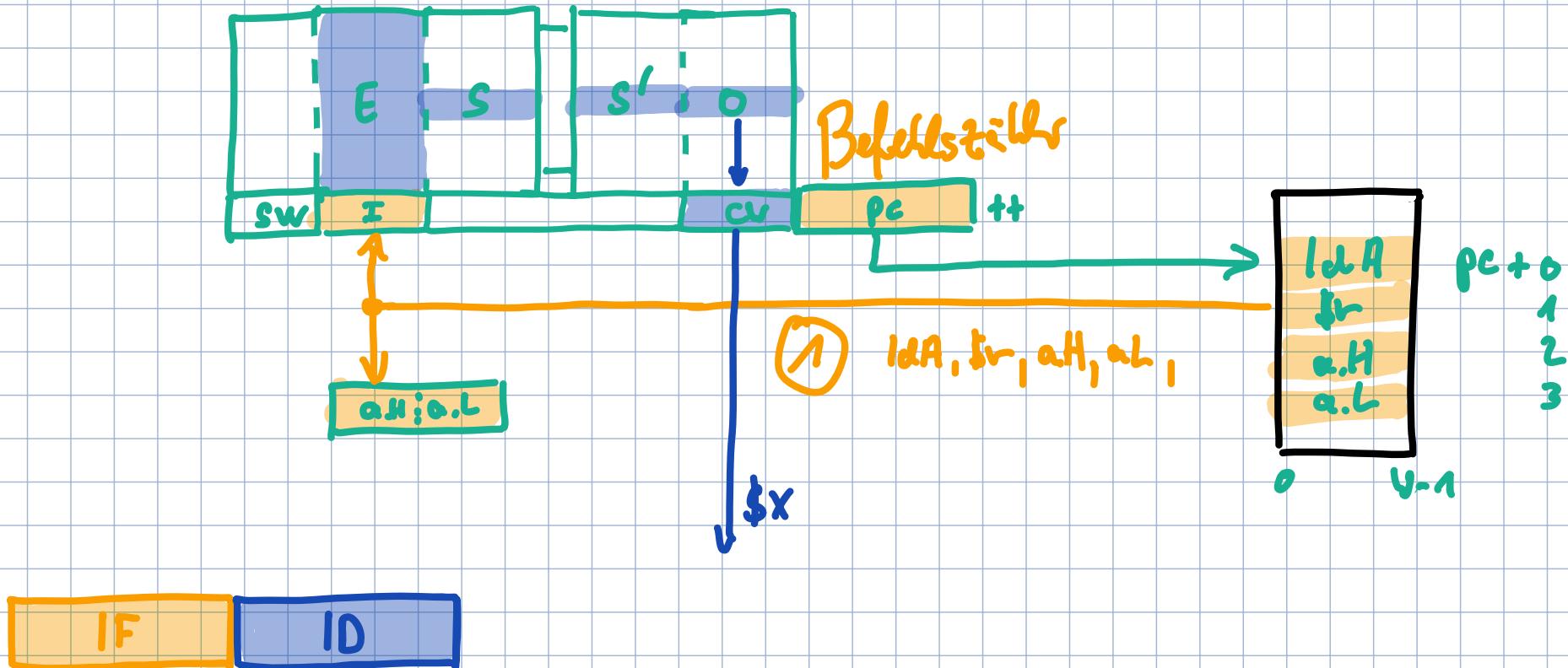


Flipboard vocabulary

Befehlszyklus : addA (a, \$X)

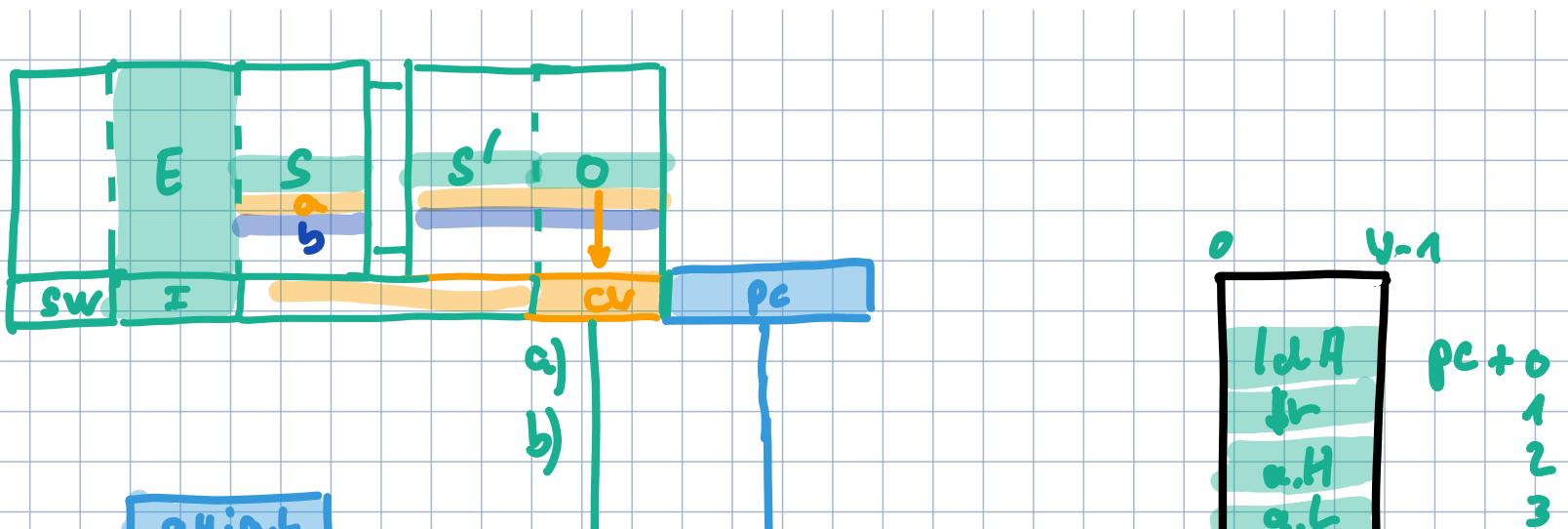
① Befehl holen (Instruction Fetch)

② Befehl dekodieren (Instruction Decode)



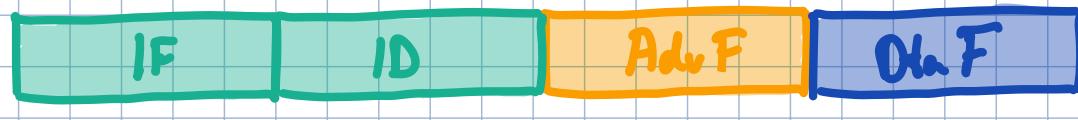
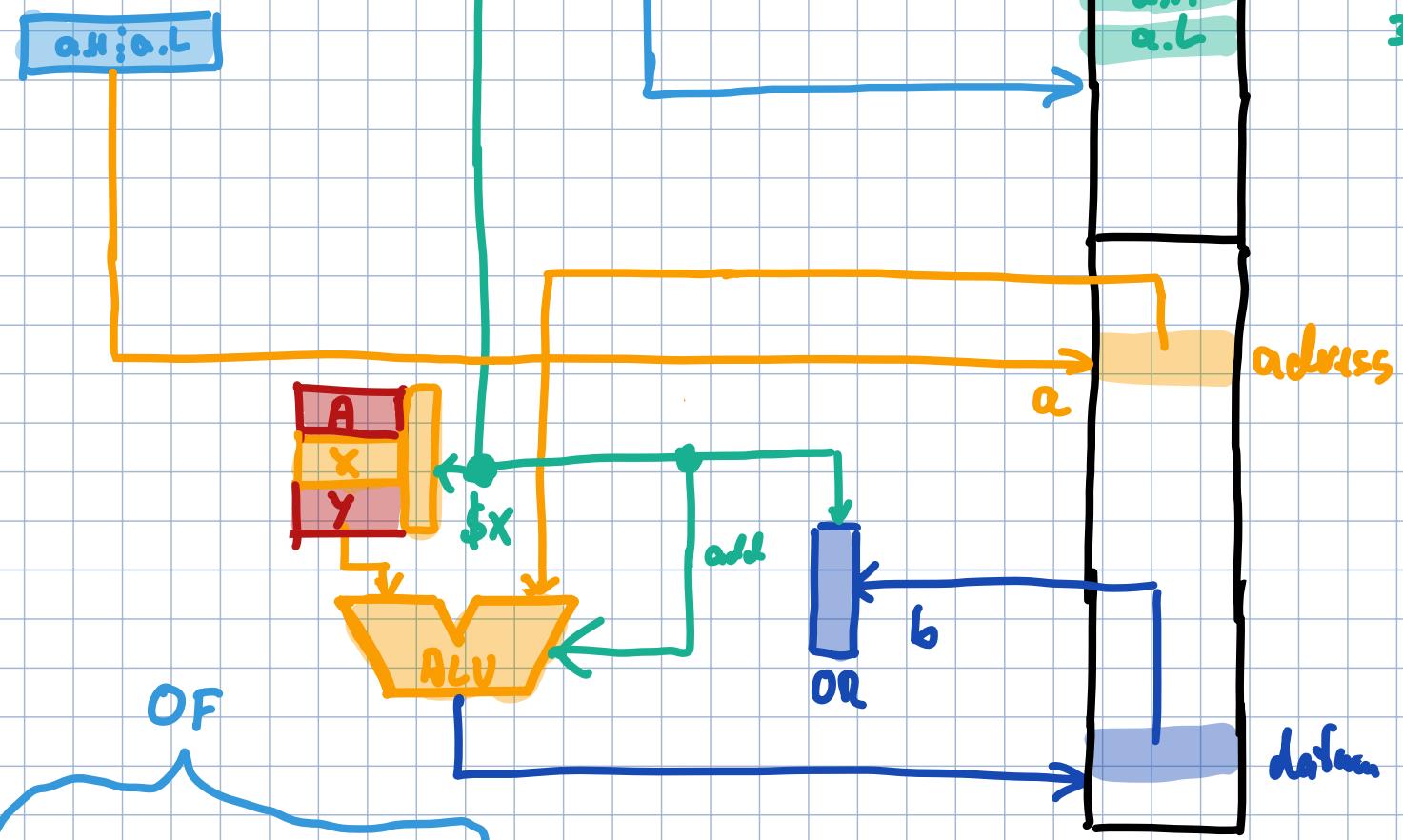
Befehlszyklus:

addA ($a, \$X$)



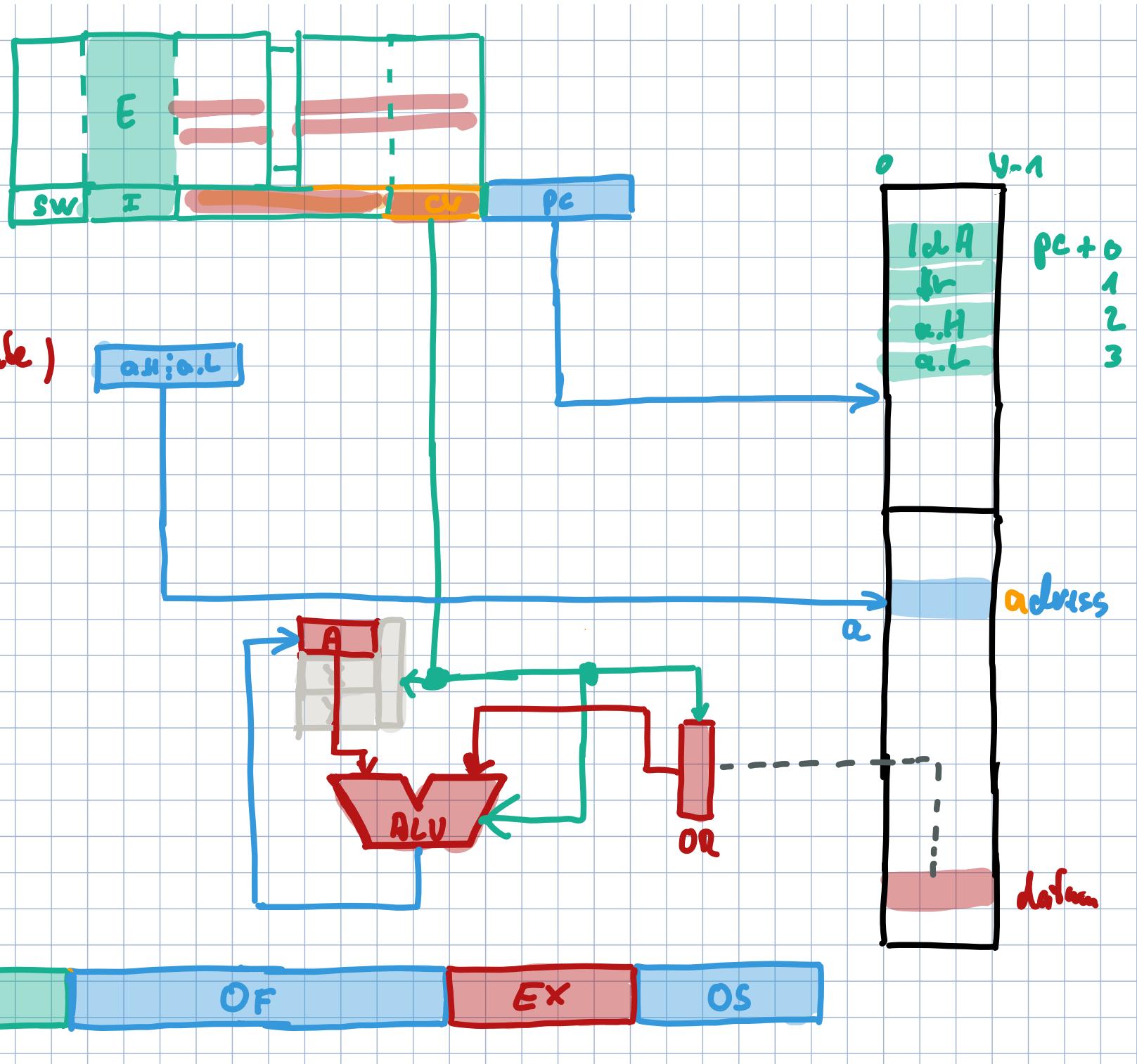
③ Operand Fetch

- a) Adresse holen
- b) Daten holen



Befehlszyklus :

addA ($a, \$X$)



Zusammenfassung: V. Membrane - Architektur

- Program und Data in gleich Sprache
- Befehlszyklus gliedert sich in vier Phasen

| | |
|--------------------|----|
| Instruction Fetch | IF |
| Instruction Decode | ID |
| Oprand Fetch | OF |
| Execute | Ex |
| Oprand Store | OS |

Befehlszyklus:

Zyklus n



n+1

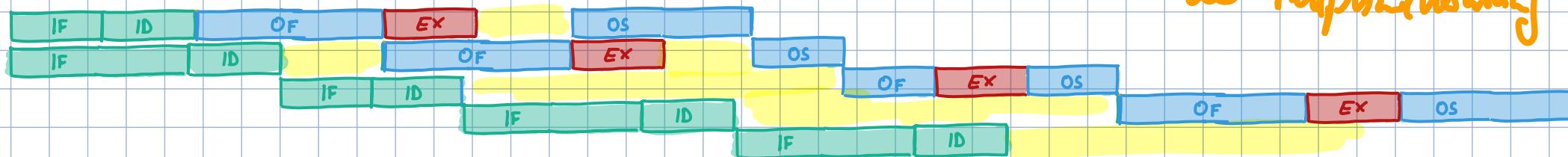


Fließbandverarbeitung: Überlappende Befehlausführung

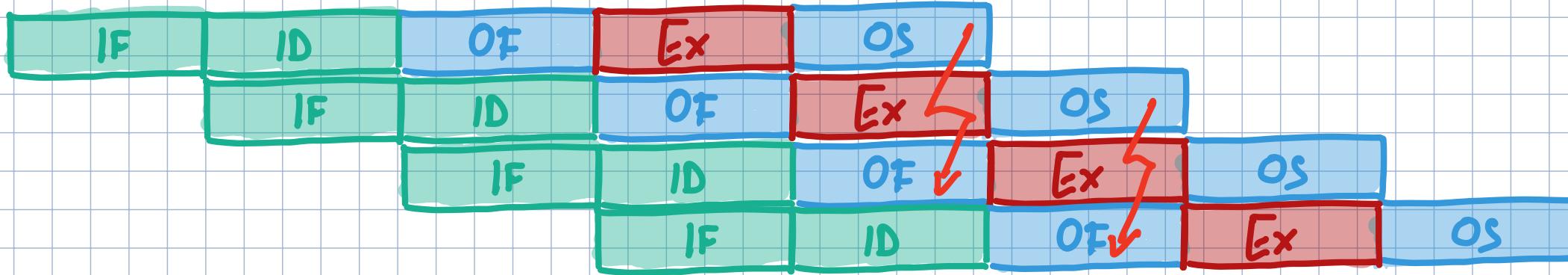
Operand Fetch und

Operand Store auf der gleichen Spritze!

Durch unterschiedliche Längen der Befehle passen unterschiedliche Größen in der Fließbandverarbeitung



Lösy: Alle Phasen deszyklus müssen gleich lang sein!



Aber: Gleichzeitiger Zugriff auf die Speicher nicht möglich

↳ Widerspruch zu v. Neumann Prinzip

Lösy: 3 - Adressmaschine mit grosser Anzahl von Registern

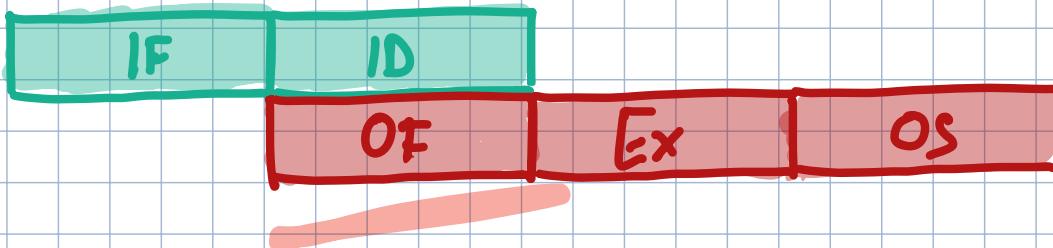
Arithmetische - Logische Operationen greifen nur auf Registerdaten zu

Hauptspeicher - Adressleiter : Progen und Daten sind getrennt.

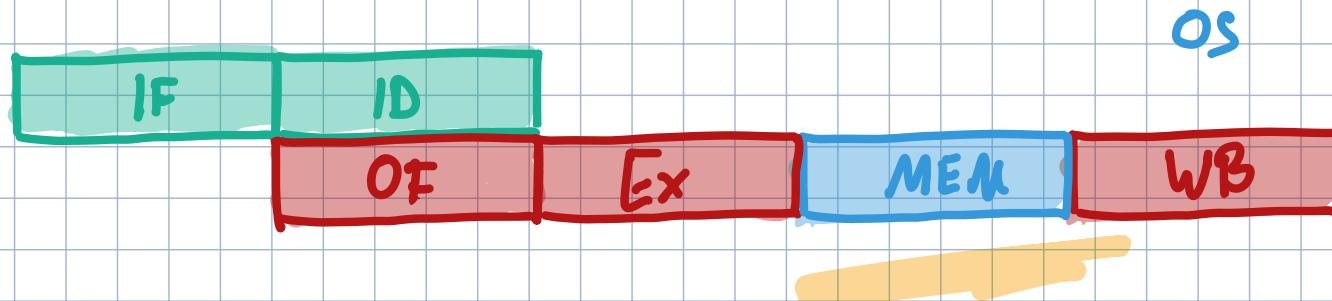
Speicherzugriff nur über
Transportleiter :

Load - Store Architektur

Load - Store Architekturen: Fließbandverarbeitung



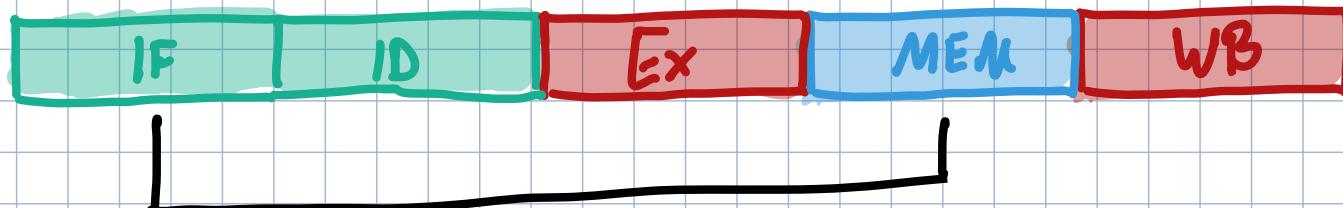
Während Operateur dekomponiert wird, kann schon auf die Register Zugriff reichen.



Transport before

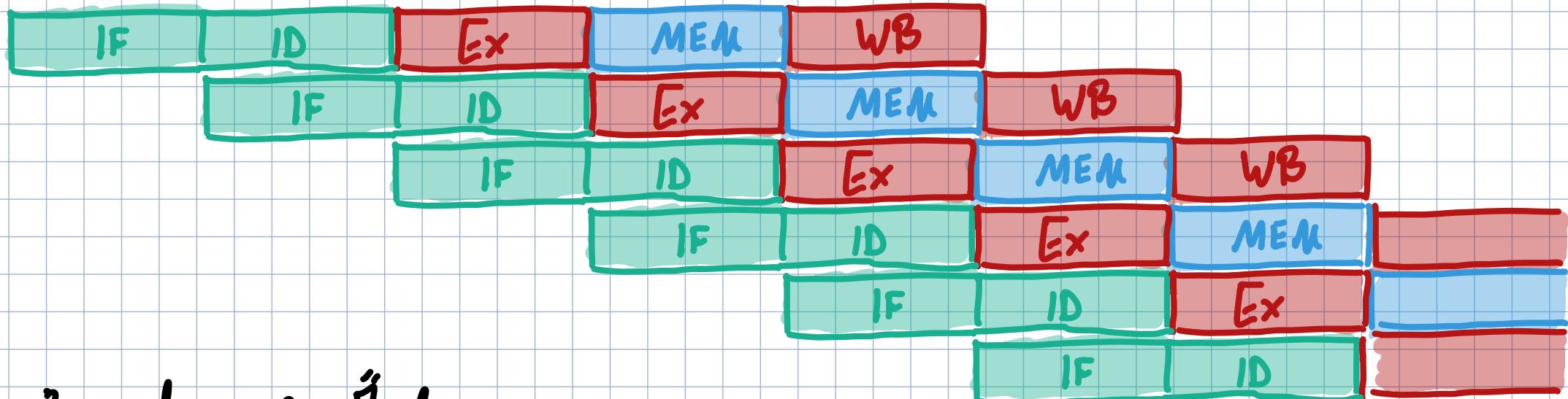
LOAD & STORE

mehr eigen Spezialisierung
notwendig



Überlappung macht Fließband-Architektur notwendig!

Flipfachadverwaltung:



Zusammenfassung der Ähnlichkeiten:

- Röhrenarquitettur - Logische Befehle ausführen unabhängig von den Registern
- Zugriff auf die Speicher wird mittels zwei Transportbefehl (LOAD / STORE)
- Begrenzte Anzahl von Adressierungssachen.
- Einheitliche Befehlsgröße

Übersetzung Complex Instruction Set Computer in
Reduced Instruction Set Computer

Beispiel Befehlssatz

Addressingarten

Befehl

add
sub
and
nor
or
sll
srl
nop
lw
st
lwv
rs
rf
bne
bog
bfg
jsw
rts

RISC

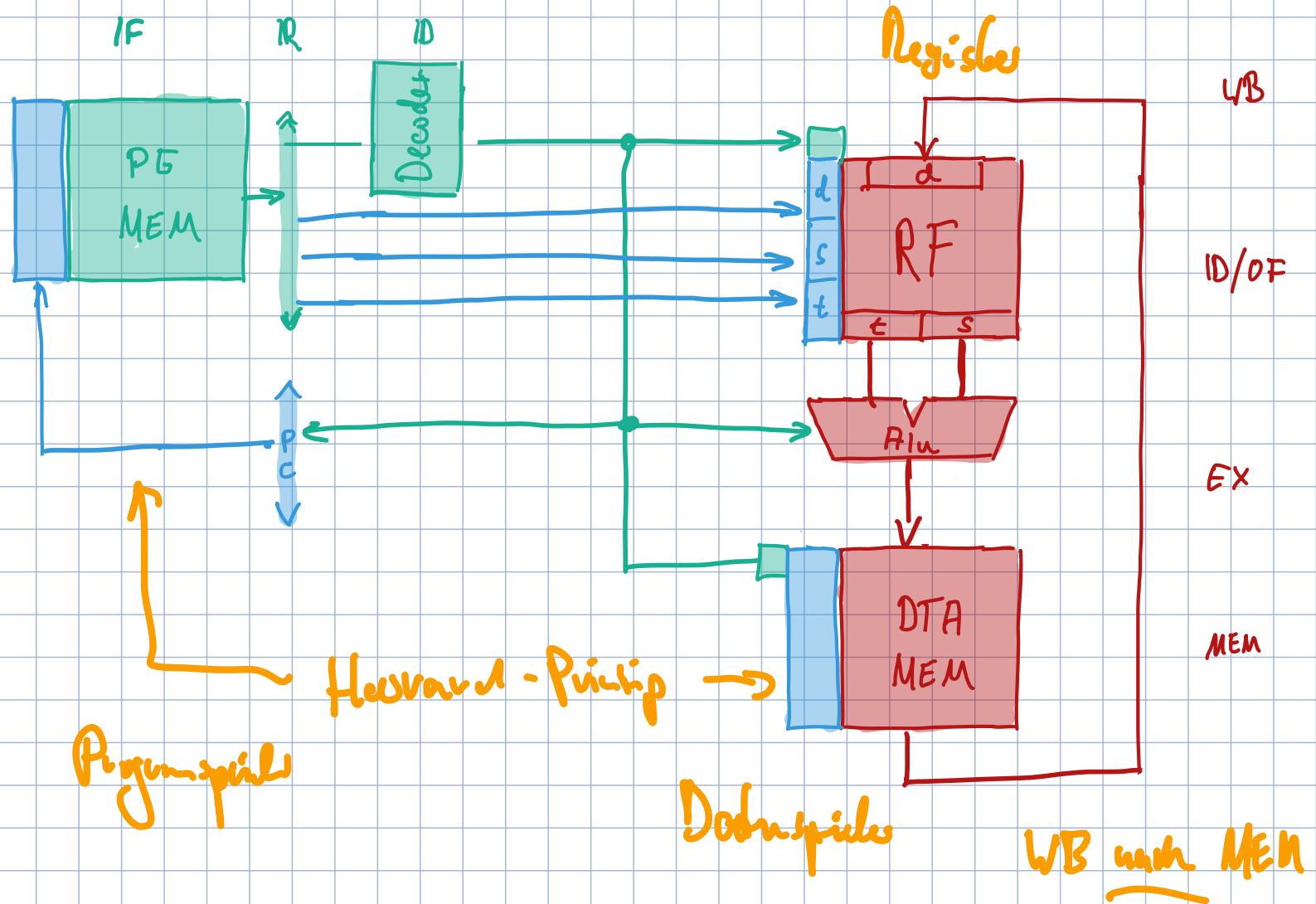
H U D X I R XI IX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |

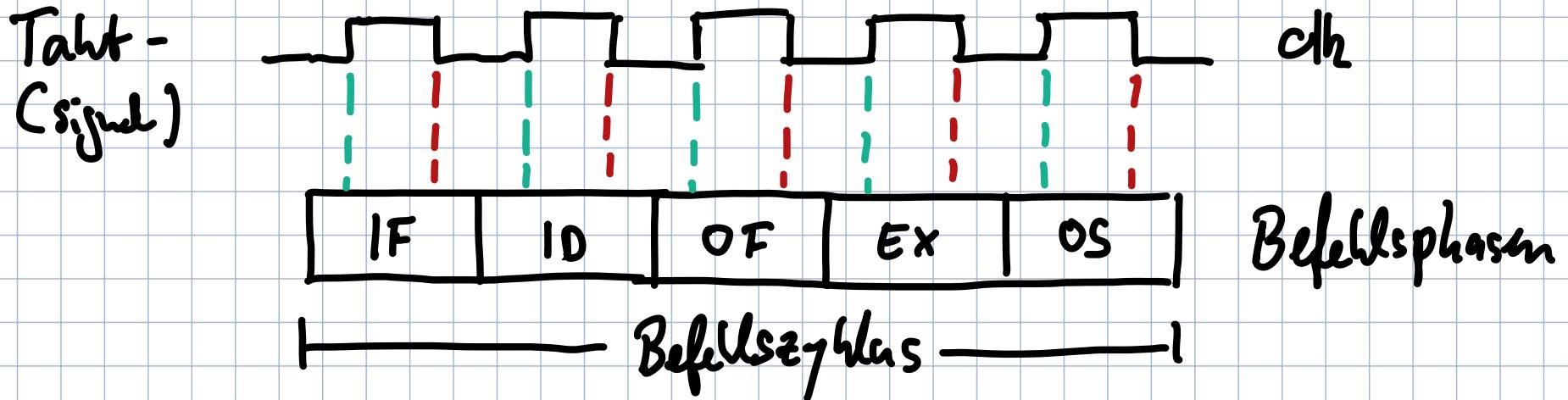
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |

X X X X X

Vereinfachte Mikroarchitektur : Harvard mit LOAD/STORE-Risikop



Das Rechen- und Steuerwach : Timing in der Mikroarchitektur



Komplexität beim Entwurf:

Never change a working system

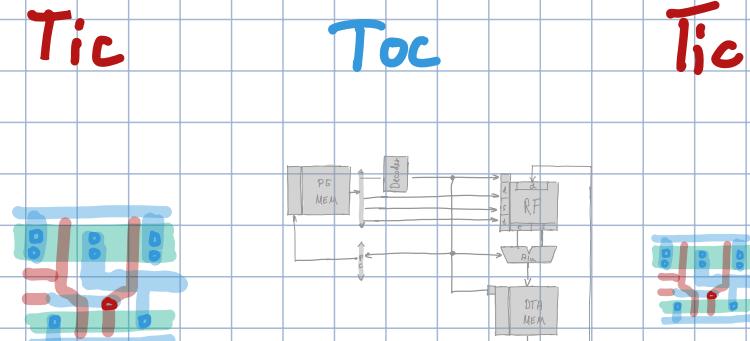
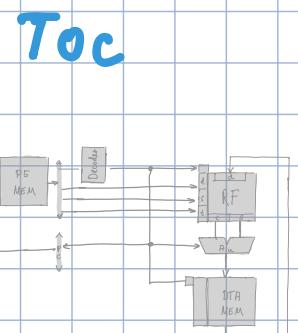
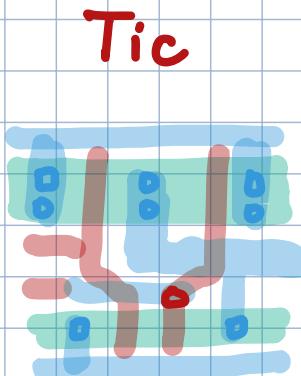


Moore's Law

Durch Moores Vorhersage nimmt jeder in der Industrie an, dass 18 Monate bis zur nächsten, schnelleren Generation vorliegen.

Daher: Fließband in der Produktions!

Technologie Veränderung der
- Strukturierung:
Architektur:



Tic