

OA (Operandenadressregister): enthält die Adresse eines der aktuellen Operanden, veränderbar durch laden vom DBI

Bezeichnung Rechenwerk für ALE, ORi, PSR und OA

BR (Befehlsregister): enthält den aktuellen Maschinenbefehl

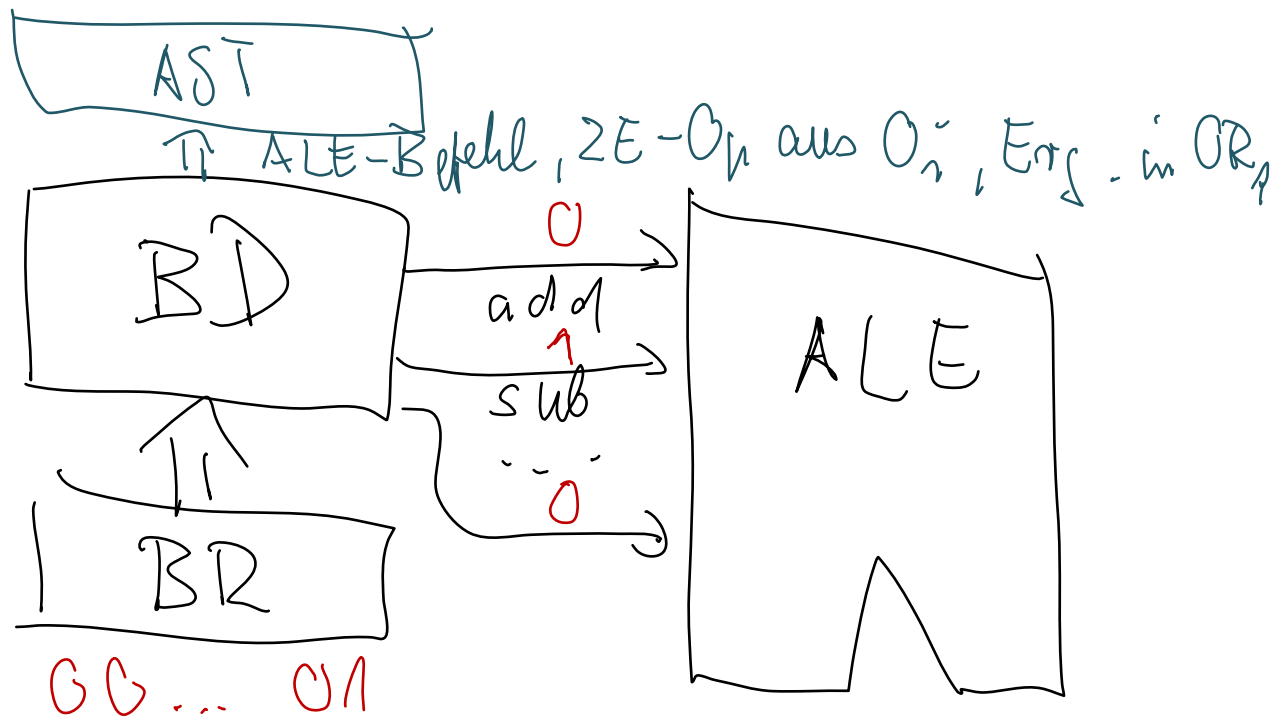
BA: Befehlsadressregister, enthält die Adresse des aktuellen Operanden

BD: Befehlsdekoeder erzeugt Steuersingnale in Abhängigkeit des aktuellen Maschinenbefehls für alle übrigen Blöcke, besonders die für die Operation zur ALE und zur AST die Signale der Befehlsvariante

Bsp. zur Codierung
Signale add, sub, ... zur ALE

Befhlcodes für

ADD	00...00
SUB	00...01
MUL	00...10
DIV	00...11
...	



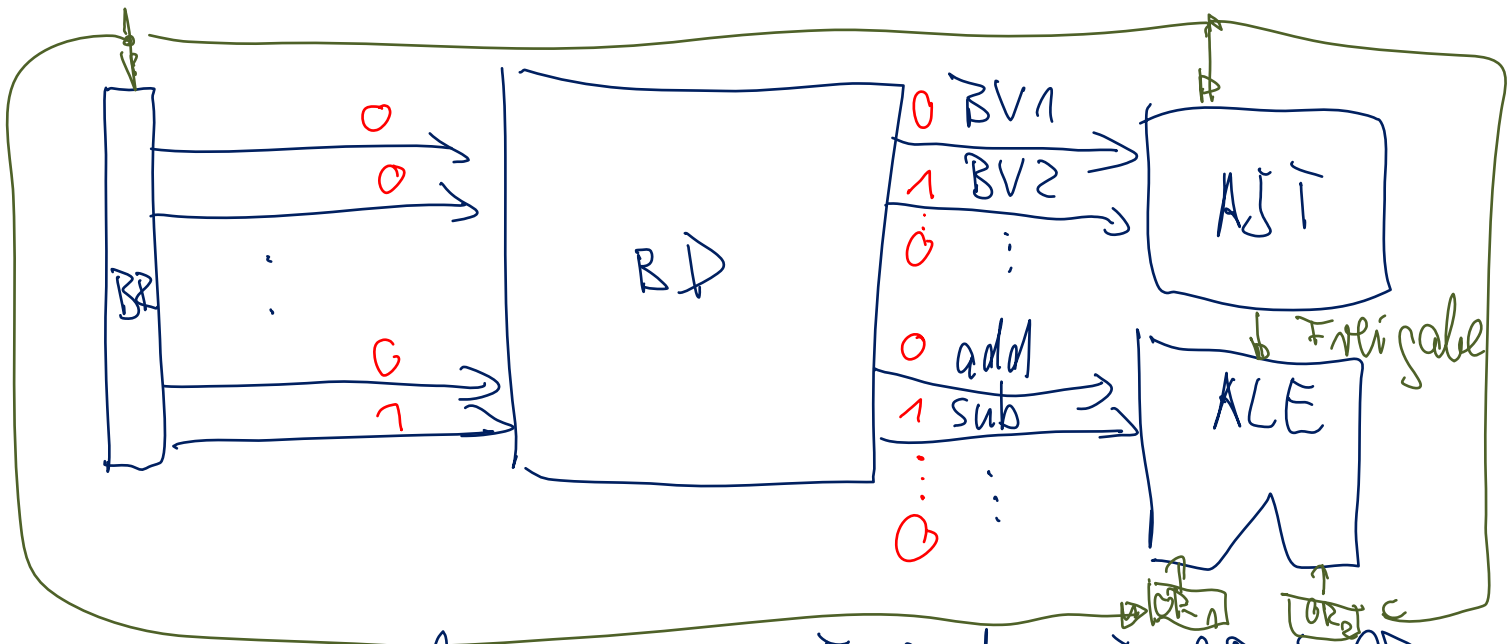
AST (Ablaufsteuerung): steuert logisch-zeitlichen Ablauf des Befehls, Signale zu allen Blöcken, vom BD die Befehlsvariante

Bsp:
SUB OR1,OR2

BD: (komplexe) kombinatorische Logik

AST: sequentieller Automat, typ. synchron (Taktbezogen)

Bsp einstufiger Dekoder:



BV2 = 1 für BV ALE-Bef. 2E-OR in OR1, Err 8. in OR1

SUB OR1, OR2

Schalttafel des Bef. Deb

(E:) BR₁₅ ... BR₂ BR₀

(A:) BV_k - BV₂ BV₁ BV₀ | add sub mul
→ ALE → ALE

ADD OR ₁ OR ₂	...	0	0
SUB OR ₁ OR ₂	...	0	1

0	...	1	0	0	1	0	0
0	...	1	0	0	0	1	0
BV gleich für beide Bef					Operation unterschiedlich		
0	...	0	1	0	0	1	0

SUB OR₁, (OA) ... 1 ... 0 1

BV ALE Bef EOP_n. 1 aus OR₁, EOP_n 2 aus

Speicher über OR₂ und
Adr. aus OA₁
Erg. in OR₂
→ BV₀ = 1

AST : PW-Modell
(Ausgangspunkt
für BV₀)
in F 3_60

Steuerwerk: BR, BA, BD und AST

elektrisches Interface: Begründung siehe weiter vorn

Datenbus (DB) bidirektional, tristate, Grund Daten vom und zum Prozessor

Adressbus (AB) unidirektional, Grund Adressen nur vom Prozessor

Steuerbus (SB) Einzelsignale unidirektional, einige vom Prozessor, die anderen zum Prozessor

elektrisches Interface: DT, AT, ST (Adress-, Daten, Steuertreiber)

3.2.2. Ablauf eines Befehls in der Grundstruktur

F 3_50, 3_60, elearnig: RA1 -> Befehlabläufe
Befehl wie oben

SUB OR1, (OA)

Anfangswerte:

BA: 100h

OA: 200h

8.6.2012

weiter Befehlsablauf in der Prozessorgrundstruktur

BA, OA schon gesetzt

Befehl gesetzt (kommt als Befehlscode später vom (Programm-) Speicher: SUB

erster Operand in OR1 gesetzt: 3

Ablauf:

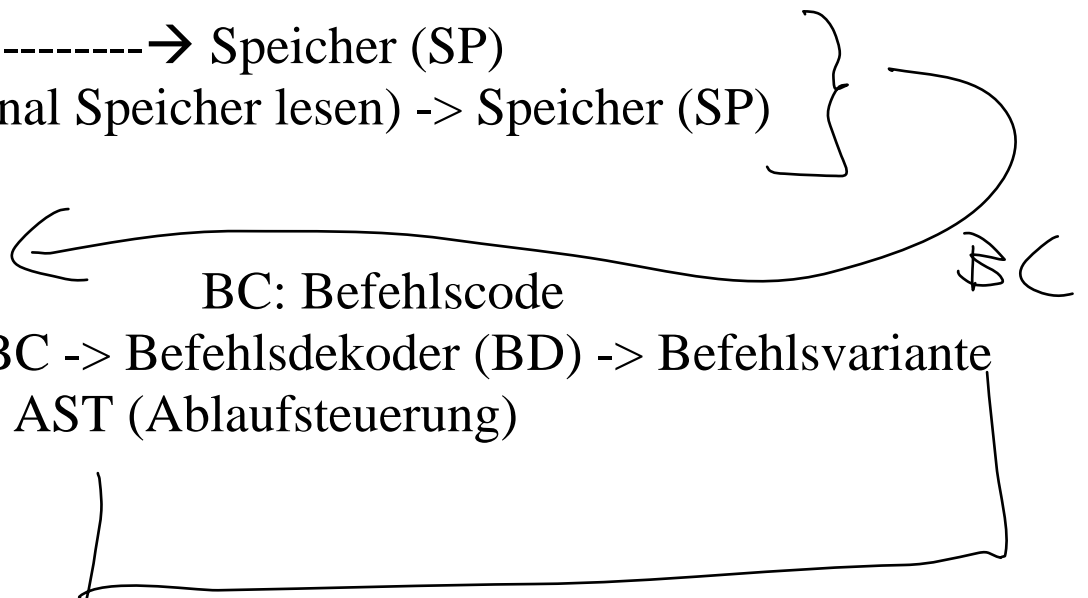
BA -> über AB -----> Speicher (SP)

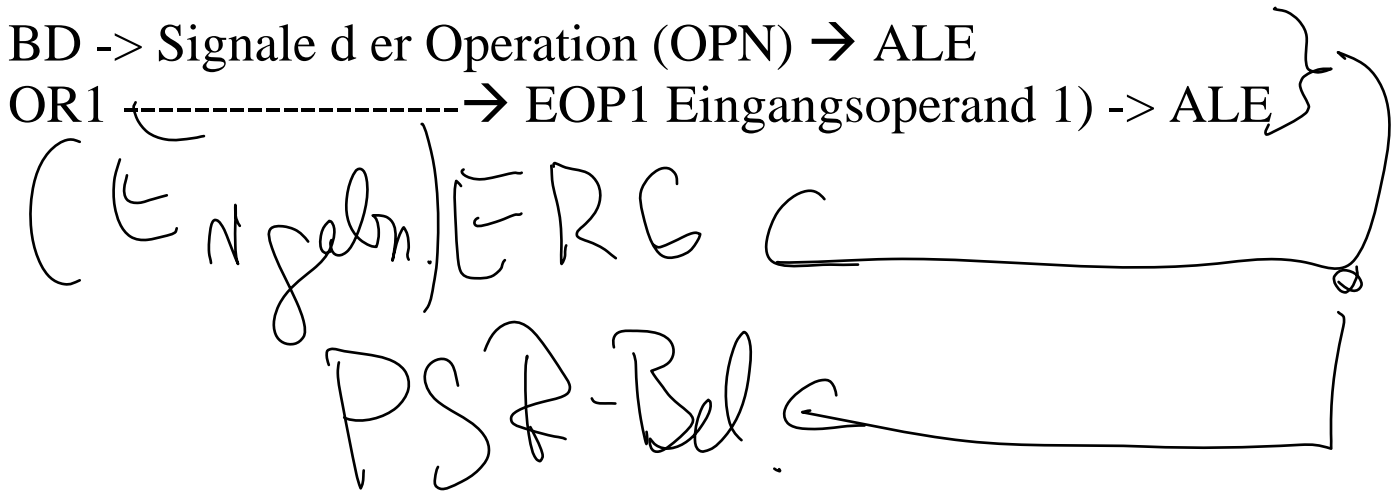
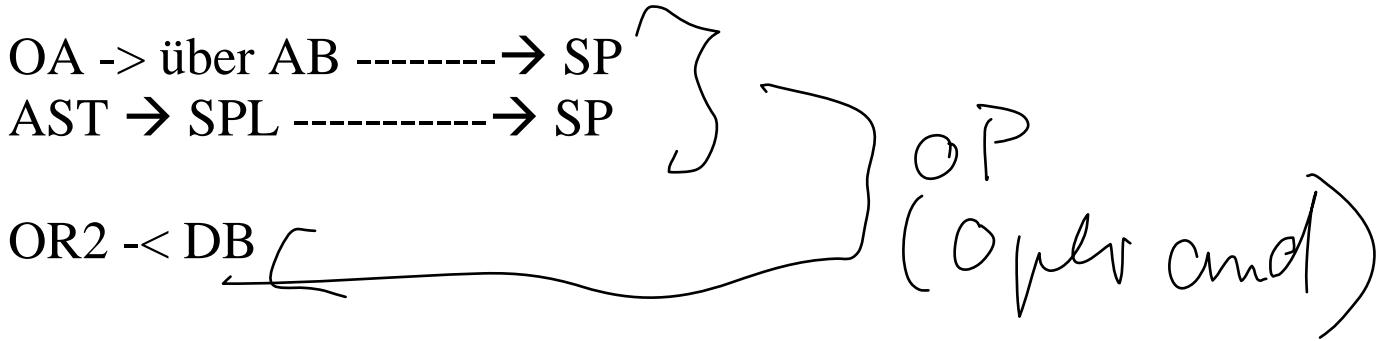
AST -> SPL (Signal Speicher lesen) -> Speicher (SP)

BR ←----- DB

BC: Befehlscode

BR → BC -> Befehlsdekoeder (BD) -> Befehlsvariante (BV) -> AST (Ablaufsteuerung)





BA -> +1 -> BA

