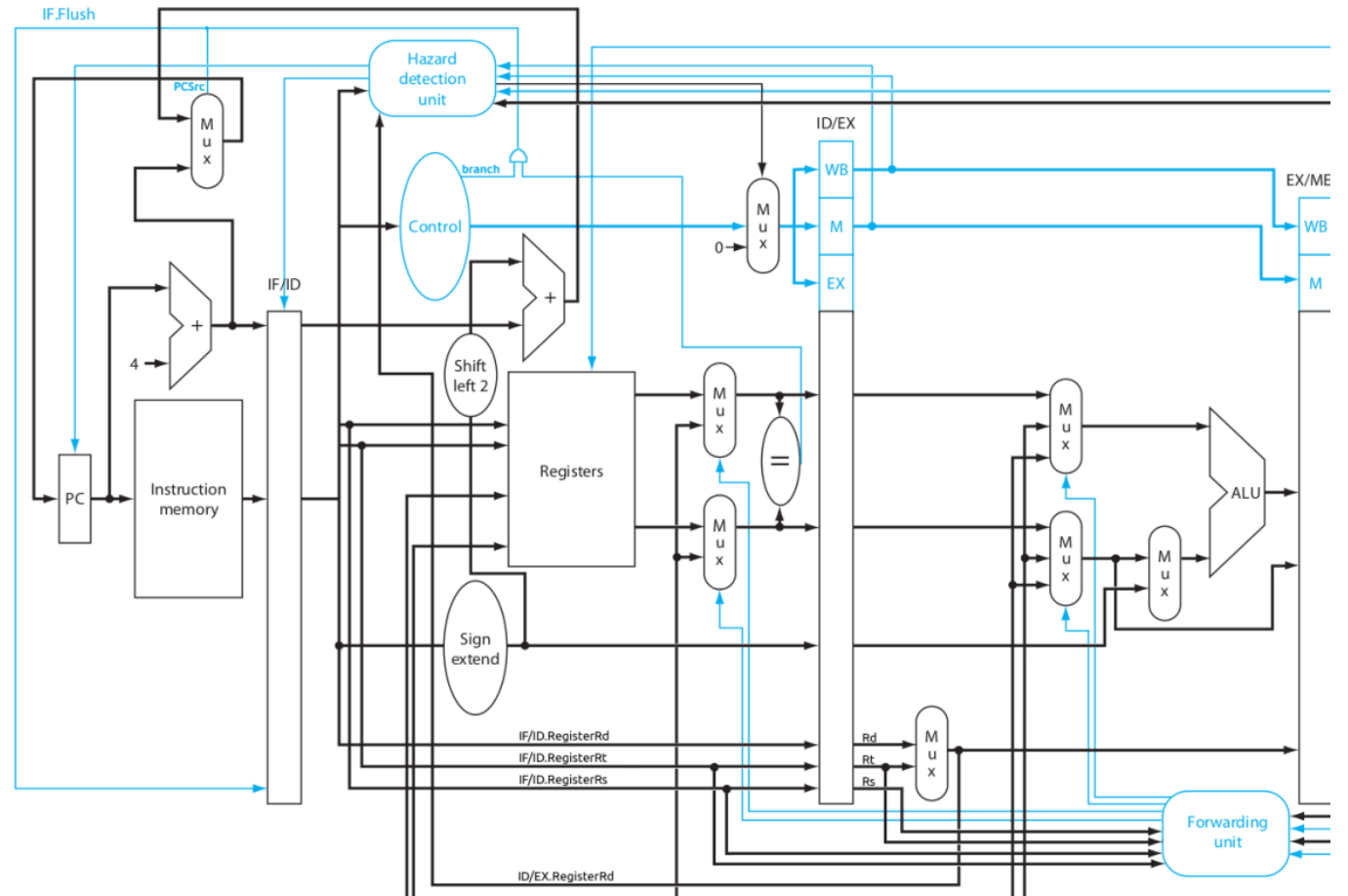
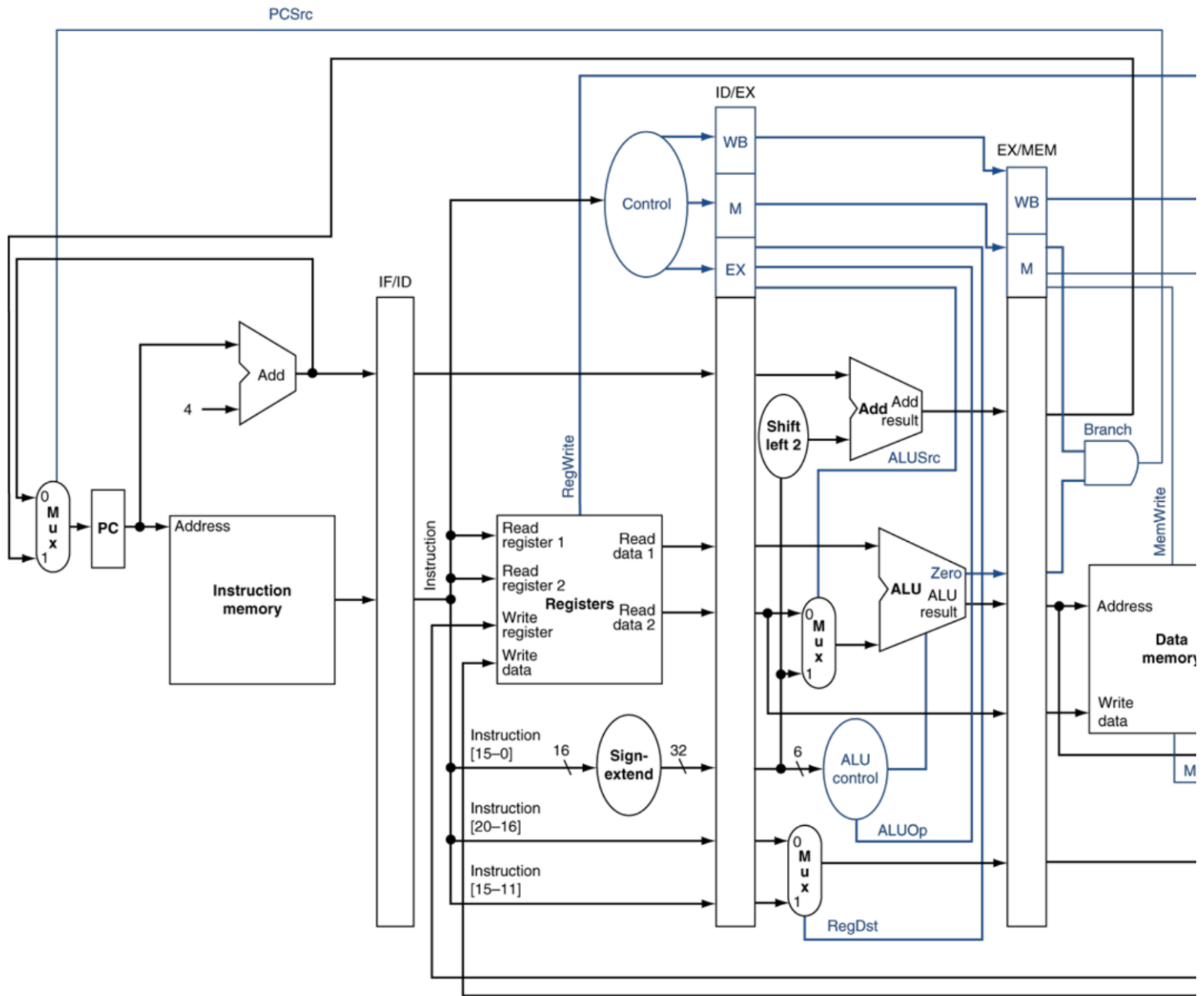


Pipelining 2

Hazard Detection
Forwarding Unit

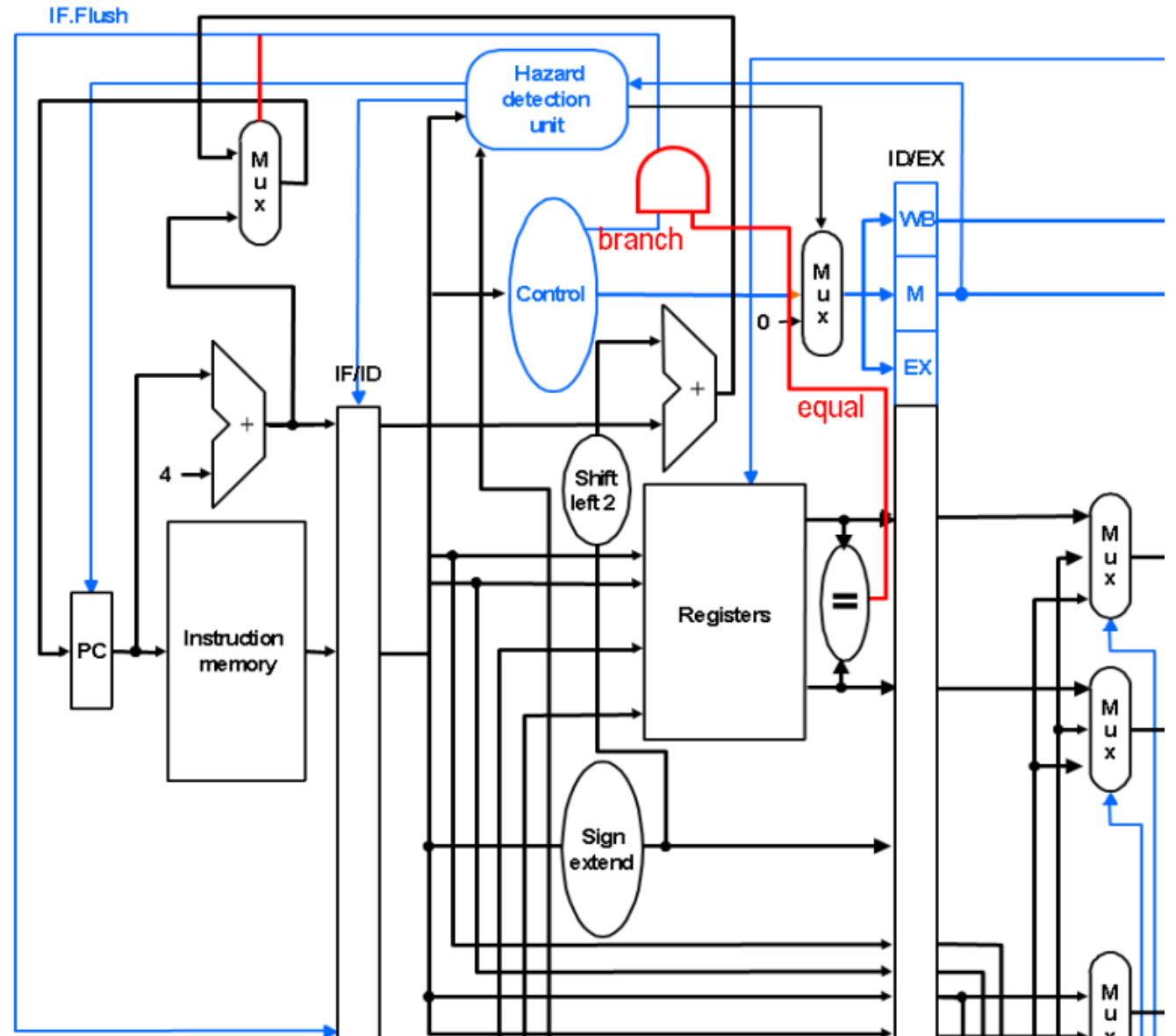


Branch letzte Woche



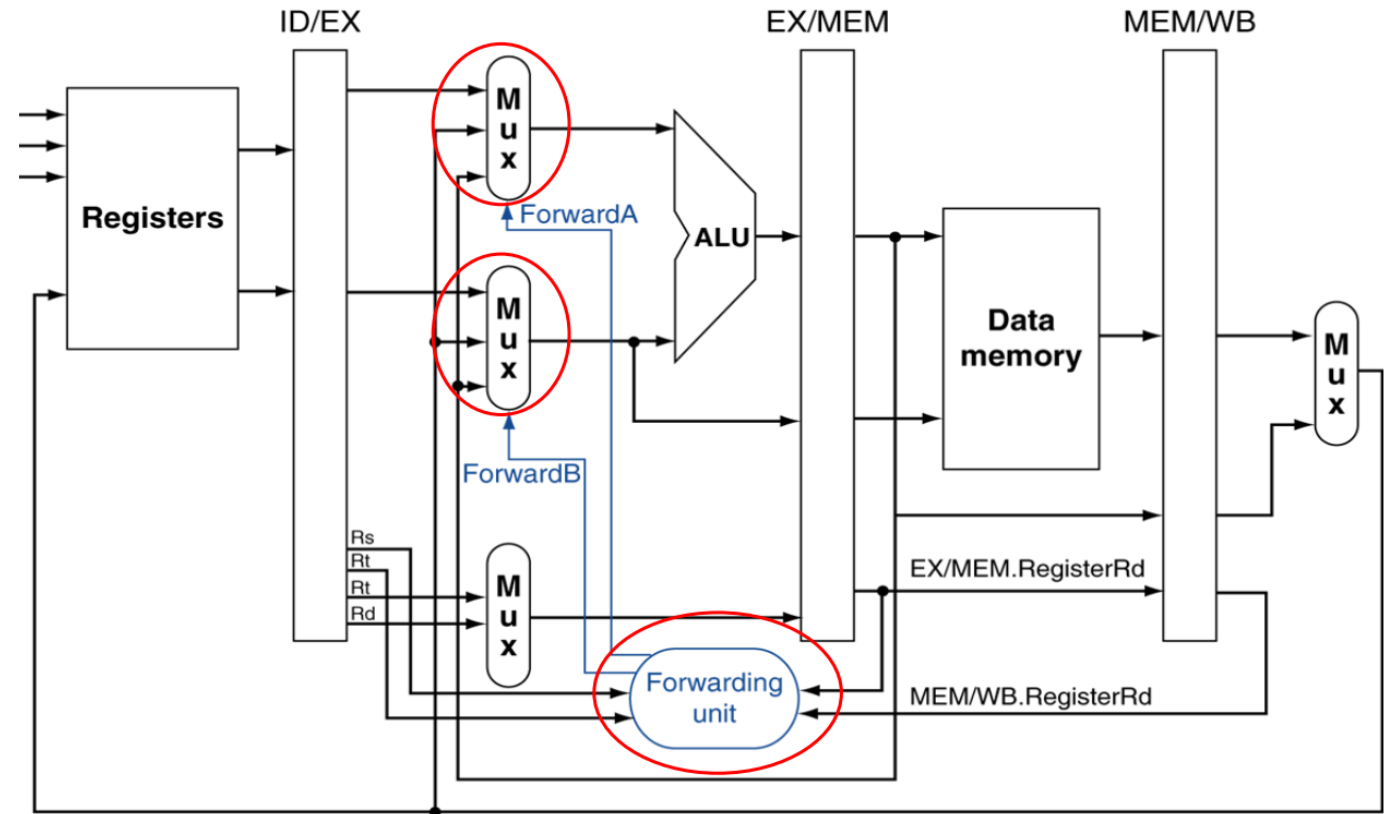
Branch in die 2. Phase (Instruction Decode)

- beq führt Vergleich von R[rs] und R[rt] in ID-Stufe aus
- Vergleich: XOR, anschließend die 32 einzelnen XOR Ergebnisse verODERN und invertieren
- Bei Gleichheit & branch = 1:
 - Sprungziel in PC schreiben
 - Falscher Befehl in Phase davor -> flushen (IF/ID auf 0 setzen)



Forwarding Unit

- Lösung Data Hazard:
 - Ergebnis aus EX oder aus MEM werden im nächsten EX benötigt
- RegWrite & Rd gleich Rt oder Rs aus EX oder ID
- Ergebnisse direkt an EX weiterleiten
 - Erst später in Register speichern
- Durch setzen von Mux
- Welches Problem bleibt?



Hazard Detection Unit

- Prüfe:
 - lw-Befehl in EX (MemRead)
 - ID.rt rs = ID/EX.rd
- Mache:
 - ControlSignale in ID auf 0 (Bubble)
 - WriteEnable = 0 für IF/ID und PC (alte Werte bleiben erhalten)
- Bis zu 2 Wartezyklen
- Lösung:
 - Load-Use-Konflikt
 - beq in ID braucht Register aus EX
 - beq in ID braucht Register aus MEM

