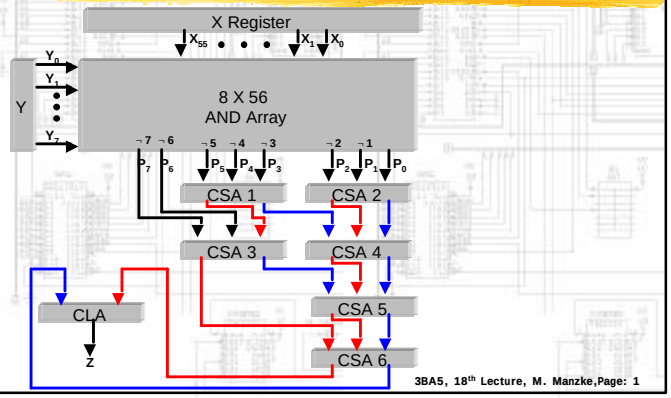
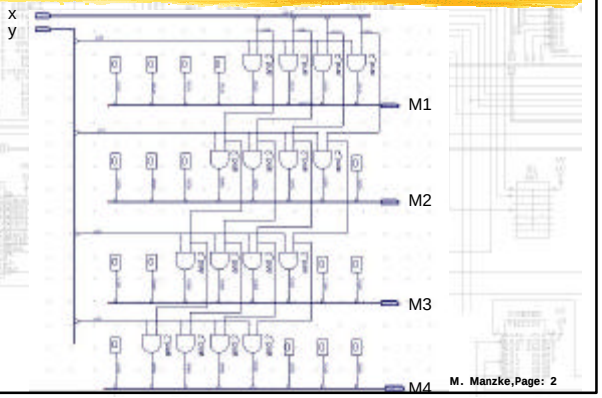
Wallace Tree Multiplier

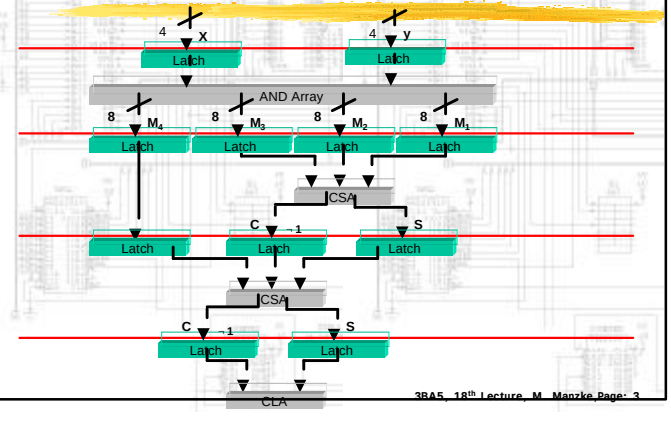
Example from Lecture (11)



AND Array (4-bit)



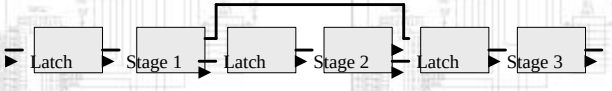
Pipelined
Wallace Tree Multiplier



Instruction Pipelines and Arithmetic Pipelines

- Can be static or dynamic.
  - Static: Must be drained before a different operation is performed
  - Dynamic: Can perform more than one operation at the same time

e.g. multiplication and addition

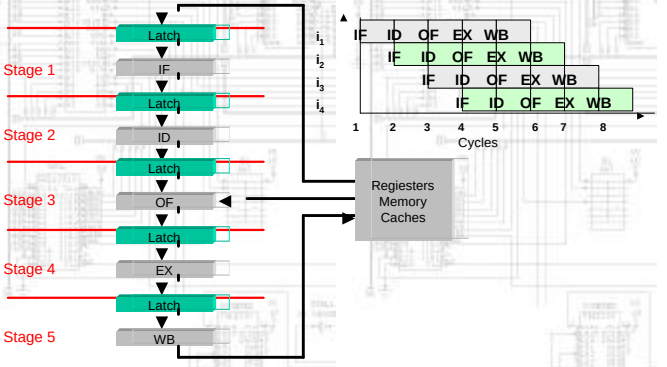


Three stage dynamic pipeline

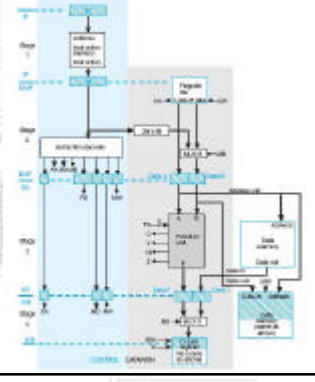
Instruction Pipelines

- The von Newmann architecture execates an instruction cycle which may be decomposed into these stages:
  - IF = Instruction fetch from cache or memory
  - ID = Instruction decode
  - OF = Operand fetch from register, cache or memory
  - EX = Apply operator to operand(s)
  - WB = Write back the results to register, cache or memory

Stages of an Instruction Pipeline



Pipelined Computer



Instruction Pipe Hazards

IF and ID

- IF - Instruction Fetch
  - An on-chip cache adjacent to the IF stage which maintains block prefetching ensures a constant supply of instructions
  - Until a branch is taken.
    - Thus branches are classified as control hazards.
- ID - Instruction Decoding
  - No hazards and complete we know:
    - What functional unit is requiered
    - Where the operand(s) reside
    - Where the results must go

Instruction Pipe Hazards
OF – Structural Hazards

- OF – Operand Fetch
  - Two classes of hazards may delay the issue of the instruction to the EX stage
    - Structural hazards

      No funtional circuit available

      eg i_1 $R_3 \leftarrow R_1 \div R_2$

      i_2 $R_6 \leftarrow R_4 \div R_5$

      Delay i_2 – EX until i_1 - WB

Instruction Pipe Hazards
OF – Data Hazards

- Data hazard (data dependency)

  Instructions which refer to results or locations which are associated with preseding but incomplete instructions
  - We must preserve the function prescribed by the program