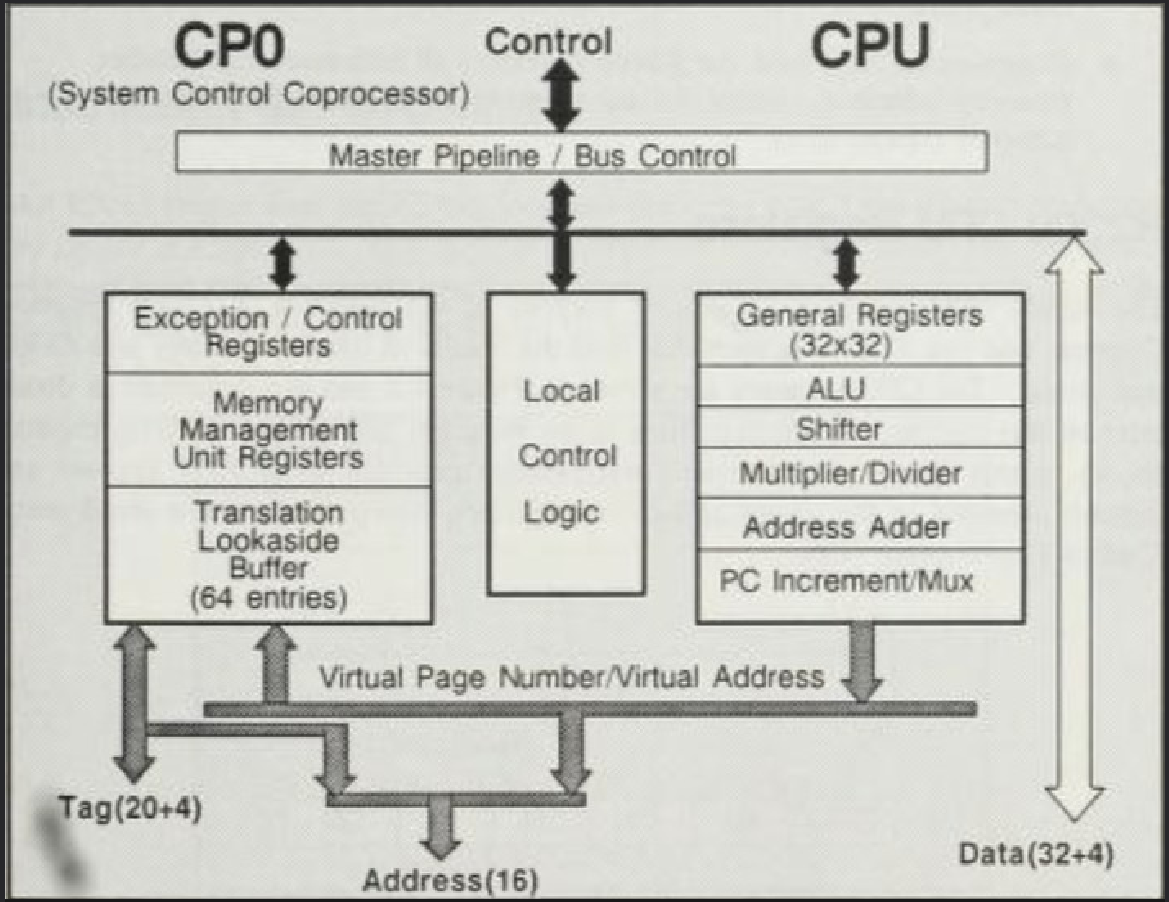
Pipeline dimensions

The depth and width of a pipeline

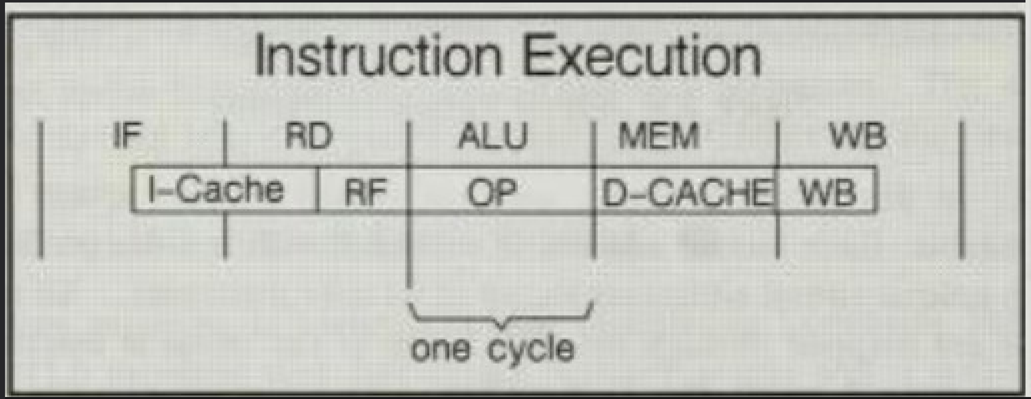
The MIPS R2000 (1986)

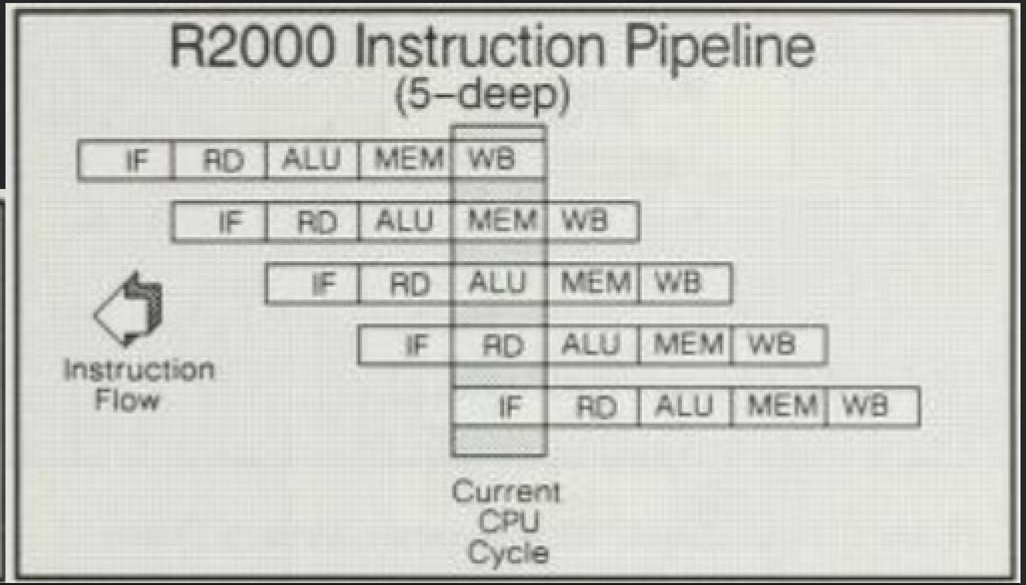

CP0
(System Control Coprocessor)
Control
CPU
Master Pipeline / Bus Control
Exception / Control Registers
Memory Management Unit Registers
Translation Lookaside Buffer (64 entries)
Local Control Logic
General Registers (32x32)
ALU
Shifter
Multiplier/Divider
Address Adder
PC Increment/Mux
Virtual Page Number/Virtual Address
Tag(20+4)
Address(16)
Data(32+4)

Kane, Gerry. MIPS RISC Architecture. Prentice-Hall, Inc., 1988.

- Coprocessor
  - Generates addresses
  - Handles all control
  - Handles exceptions
- CPU
  - Pipelined processor
  - Autonomous multiplier/divider

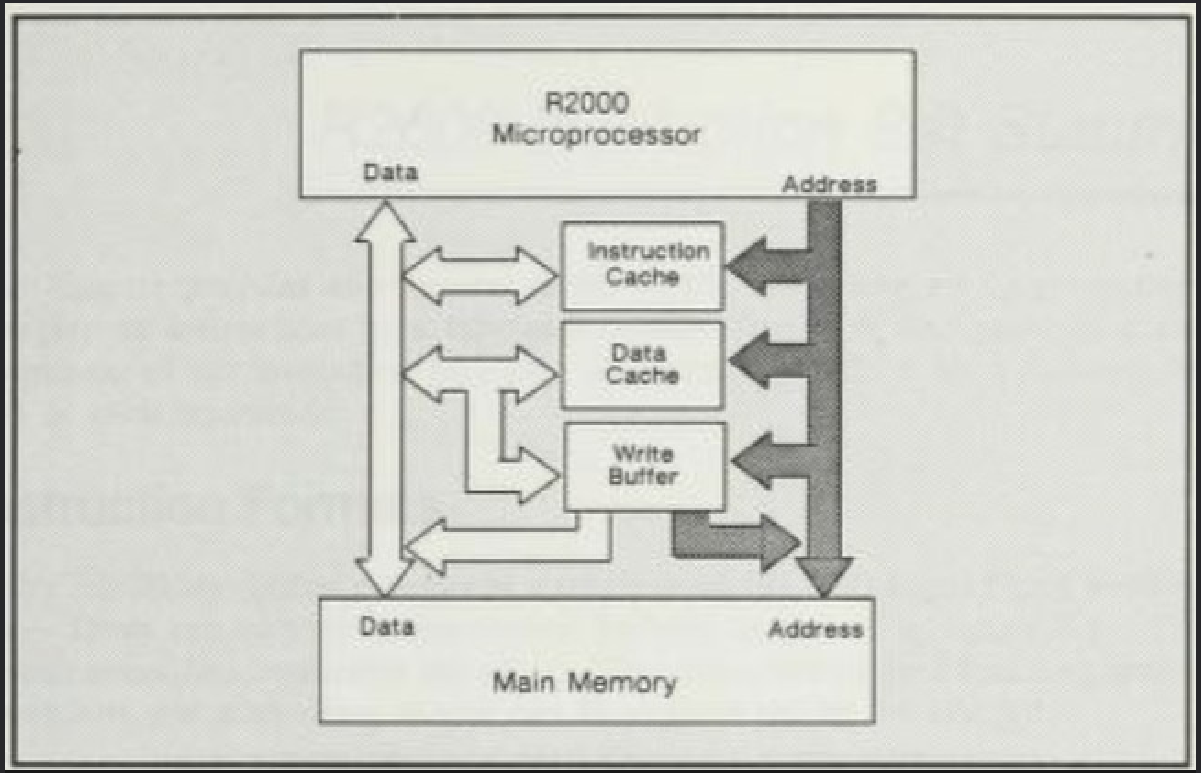
The MIPS R2000 pipeline





Kane, Gerry. MIPS RISC Architecture. Prentice-Hall, Inc., 1988.

The MIPS R2000 memory hierarchy



Kane, Gerry. MIPS RISC Architecture. Prentice-Hall, Inc., 1988.

- The caches were actually “off chip”
  - But still closer and faster to access than main memory
- Used a write-through policy to keep main memory up-to-date

How can we improve execution time?

$$Execution\ Time = \left(\frac{instructions}{program}\right)\left(\frac{cycles}{instruction}\right)\left(\frac{seconds}{cycle}\right)$$

- Reduce cycle time
  - Temporal parallelism
  - Pipeline depth
- Reduce CPI (increase IPC)
  - Spatial parallelism
  - Pipeline width

Increasing pipeline depth

Advantages

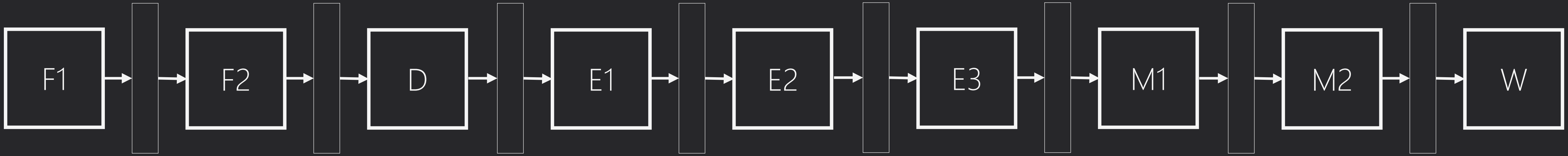
- Increases the frequency
  - Equivalently: Reduces the cycle time (T_C)

Disadvantages

- Limited by pipeline registers
  - Sequencing overhead
  - Extra hardware
- Limited by hazards
  - More complex control and hazard detection
  - More hardware for bypassing
  - Branches resolved later in the pipeline; Penalty of a flush increases

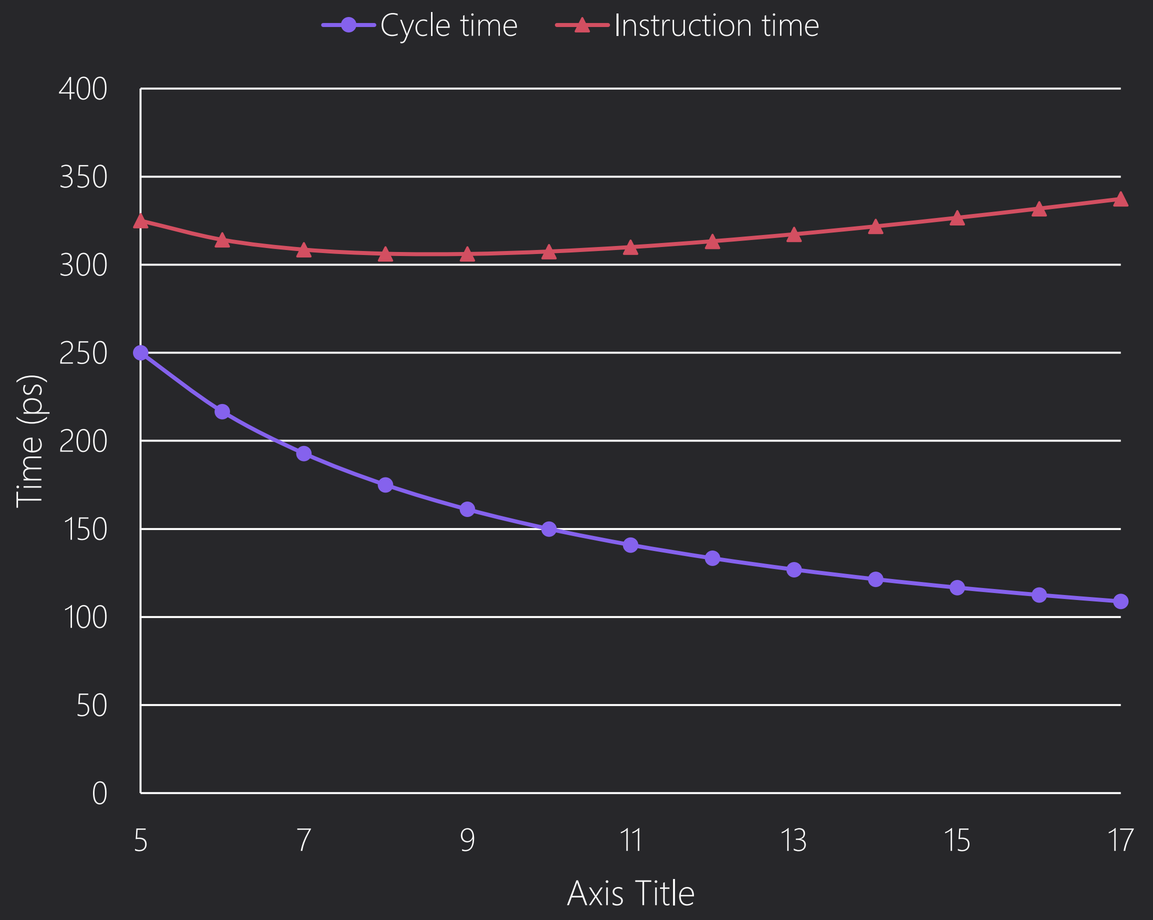
Superpipelined CPU

When a CPU has more than the classical 5 stages, we say it is *superpipelined*.



The limits of pipelining

- Suppose a single-cycle implementation can be clocked at 1 GHz (1000 ps cycle time)
- For each pipeline stage added:
  - Sequencing overhead of 50 ps
  - N = 5; $CPI_{base} = 1.3$
  - N > 5; $CPI_{penalty} = 0.15 * N$
- How does pipelining the datapath to N stages impact:
  - The cycle time (T_C)
  - The instruction time ($CPI \times T_C$)



Increasing pipeline width

Advantages

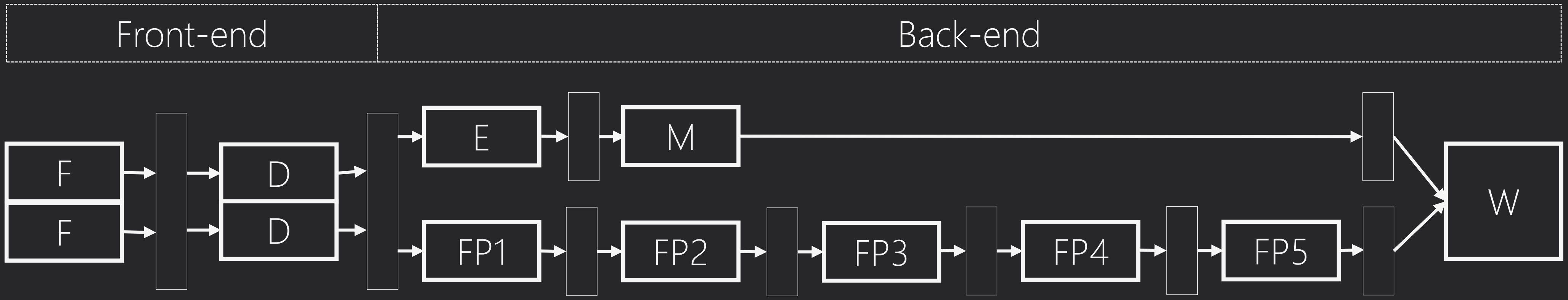
- Increases the *ideal IPC* beyond 1
  - Allows more than one instruction in a stage at a time
  - More options to exploit instruction-level parallelism

Disadvantages

- Significant changes to our existing pipeline structure
- Complex scheduling
- Increased cost
  - Area
  - Power

Superscalar CPU

- m is the (ideal) number of instructions fetched into the pipeline
- If m = 1, the processor is simply *scalar*
- m-way superscalar; here m = 2



The limits of superscalar

- Suppose an m-way superscalar with a frequency k times that of a scalar processor
- Theoretical speedup: $m \times k$

- Large m needs lots of ILP
  - Limits to hardware discovering ILP
  - Limits to ILP
- Back-end needs at least m functional units
  - Number of forwarding paths grows quadratically with number of functional units
- More ports are costly
  - e.g., register file needs more read ports
- Power
  - Increasing m and k makes for a very power-hungry processor

Conclusion

Recap of important points

Improving execution time

Pipeline Dimension	Improvement	Trade-off
Depth	Better frequency (lower cycle time)	Diminishing returns due to: • Sequencing overhead • Hazards
Width	Better ideal IPC ($IPC_{ideal} = m$)	Diminishing returns due to: • Limits of ILP and scheduling • Hardware cost