

CPU Performance

Ingo Sander
ingo@imit.kth.se



© 2000 Wolf (Morgan Kaufman)

Elements of CPU performance

- Cycle time.
- CPU pipeline.
- Memory system.



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

2

Pipelining

- Several instructions are executed simultaneously at different stages of completion.
- Various conditions can cause **pipeline bubbles** that reduce utilization:
 - branches;
 - memory system delays;
 - etc.



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

3

Pipeline structures

- Both ARM 7 and SHARC have 3-stage pipes:
 - **fetch** instruction from memory;
 - **decode** opcode and operands;
 - **execute**.



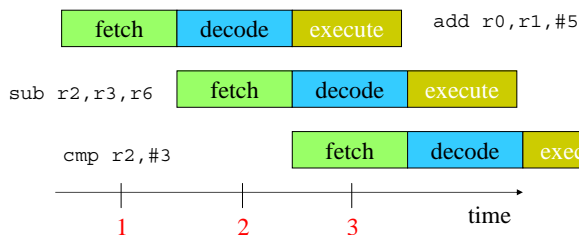
© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

4

ARM pipeline execution



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

5

Performance measures

- **Latency**: time it takes for an instruction to get through the pipeline.
- **Throughput**: number of instructions executed per time period.
- Pipelining increases throughput without reducing latency.



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

6



Pipeline stalls

- If every step cannot be completed in the same amount of time, pipeline stalls.
- Bubbles introduced by stall increase latency, reduce throughput.

© 2000 Wolf (Morgan Kaufman)

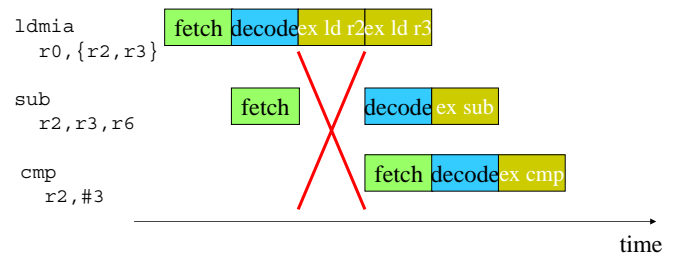
September 10, 2004

2B1447 Embedded Systems

7



ARM multi-cycle LDMIA instruction



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

8



Control stalls

- Branches often introduce stalls (branch penalty).
 - Stall time may depend on whether branch is taken.
- May have to squash instructions that already started executing.
- Don't know what to fetch until condition is evaluated.

© 2000 Wolf (Morgan Kaufman)

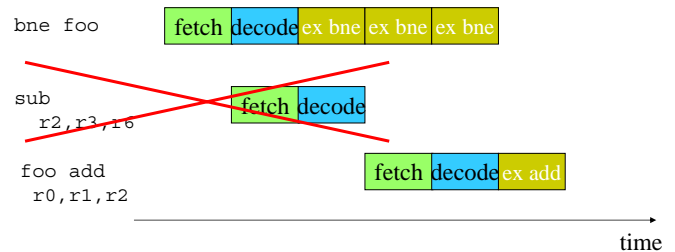
September 10, 2004

2B1447 Embedded Systems

9



ARM pipelined branch



© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

10



Example: ARM execution time

- Determine execution time of FIR filter:


```
for (i=0; i<N; i++)
  f = f + c[i]*x[i];
```
- Only branch in loop test may take more than one cycle.
 - BLT loop takes 1 cycle best case, 3 worst case.

© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

11



Superscalar execution

- Superscalar processor can execute several instructions per cycle.
 - Uses multiple pipelined data paths.
- Programs execute faster, but it is harder to determine how much faster.

© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

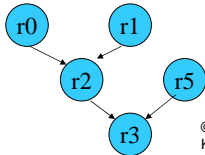
12



Data dependencies

- Execution time depends on operands, not just opcode.
- Superscalar CPU checks data dependencies dynamically:

data dependency
 add r2, r0, r1
 add r3, r2, r5



© 2000 Wolf (Morgan Kaufman)



Memory system performance

- Caches introduce indeterminacy in execution time.
 - Depends on order of execution.
- **Cache miss penalty**: added time due to a cache miss.
- Several reasons for a miss: compulsory, conflict, capacity.

© 2000 Wolf (Morgan Kaufman)



CPU power consumption

- Most modern CPUs are designed with power consumption in mind to some degree.
- Power vs. energy:
 - heat depends on power consumption;
 - battery life depends on energy consumption.

© 2000 Wolf (Morgan Kaufman)



CMOS power consumption

- **Voltage drops**: power consumption proportional to V^2 .
- **Toggling**: more activity means more power.
- **Leakage**: basic circuit characteristics; can be eliminated by disconnecting power.

© 2000 Wolf (Morgan Kaufman)



CPU power-saving strategies

- Reduce power supply voltage.
- Run at lower clock frequency.
- Disable function units with control signals when not in use.
- Disconnect parts from power supply when not in use.

© 2000 Wolf (Morgan Kaufman)



Power management styles

- **Static power management**: does not depend on CPU activity.
 - Example: user-activated power-down mode.
- **Dynamic power management**: based on CPU activity.
 - Example: disabling off function units.

© 2000 Wolf (Morgan Kaufman)

Application: PowerPC 603 energy features



- Provides doze, nap, sleep modes.
- Dynamic power management features:
 - Uses static logic.
 - Can shut down unused execution units.
 - Cache organized into subarrays to minimize amount of active circuitry.

© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

19

PowerPC 603 activity



- Percentage of time units are idle for SPEC integer/floating-point:

unit	Specint92	Specfp92
D cache	29%	28%
I cache	29%	17%
load/store	35%	17%
fixed-point	38%	76%
floating-point	99%	30%
system register	89%	97%

© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

20

Power-down costs



- Going into a power-down mode costs:
 - time;
 - energy.
- Must determine if going into mode is worthwhile.
- Can model CPU power states with power state machine.

© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

21

Application: StrongARM SA-1100 power saving



- Processor takes two supplies:
 - VDD is main 3.3V supply.
 - VDDX is 1.5V.
- Three power modes:
 - Run: normal operation.
 - Idle: stops CPU clock, with logic still powered.
 - Sleep: shuts off most of chip activity; 3 steps, each about 30 μ s; wakeup takes > 10 ms.

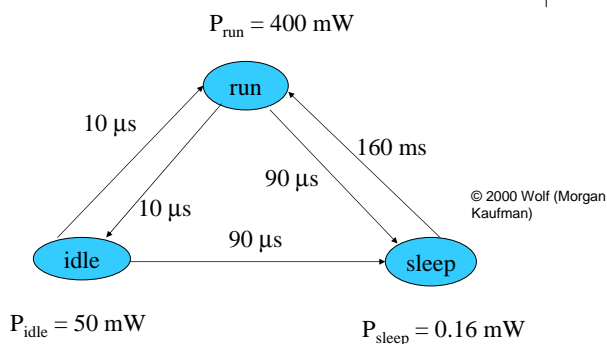
© 2000 Wolf (Morgan Kaufman)

September 10, 2004

2B1447 Embedded Systems

22

SA-1100 power state machine



September 10, 2004

2B1447 Embedded Systems

23