ELEC3020: Lecture 2-2

Early iA32 processors
386 and 486

With thanks to CS370, Superscalar Processing at CMU

Superscalar Processing

- Intel Processors
  - 486, Pentium, Pentium Pro etc.
- Superscalar Examples
  - PowerPC 604
  - MIPS R10000
  - DEC Alpha 21264
  - Speculative Execution, Register Renaming, Branch Prediction

Intel x86 Processors

Processor	Year	Transistors	MHz	Spec92 (Int/FP)	Spec95 (Int/FP)
8086	’78	29K	4	*Basis of IBM PC & PC-XT*	
i286	’83	134K	8	*Basis of IBM PC-AT*	
i386	‘86	275K	16		
	’88		33	6 / 3	
i486	‘89	1.2M	20		
			50	28 / 13	
Pentium	‘93	3.1M	66	78 / 64	
			150	181 / 125	4.3 / 3.0
PentiumPro	‘95	5.5M	150	245 / 220	6.1 / 4.8
			200	320 / 283	8.2 / 6.0
Pentium II	‘97	7.5M	300		11.6 / 6.8

Other Processors (same generation)

Processor	Year	Transistors	MHz	Spec92	Spec95
MIPS R3000	’88	110k	25	16.1 / 21.7	
MIPS R5000	‘96	3.7M	180		4.1 / 4.4
MIPS R10000	‘96	6.8M	200	300 / 600	8.9 / 17.2
Alpha 21164a	‘96	9.3M	417	500 / 750	11 / 17
Alpha 21264	‘98	15M	500		30 / 60

Architectural Performance

- Metric
  - SpecX92/Mhz: Normalized with respect to clock speed
  - But…one measure of architectural quality is the clock speed for a given process.
- Sampling

Processor	MHz	SpecInt92	IntAP	SpecFP92	FltAP
i386/387	33	6	0.2	3	0.1
i486DX	50	28	0.6	13	
Pentium	150	181	1.2	125	0.8
PentiumPro	200	320	1.6	283	1.4
MIPS R3000A	25	16.1	0.6	21.7	0.9
MIPS R10000	200	300	1.5	600	3.0
Alpha 21164a	417	500	1.2	750	1.8

x86 ISA Characteristics

- Multiple Data Sizes and Addressing Methods
  - Recent generations optimized for 32-bit mode
- Limited Number of Registers
  - Stack-oriented procedure call and FP instructions
  - Programs reference memory heavily (41%)
- Variable Length Instructions
  - First few bytes describe operation and operands
  - Remaining ones give immediate data & address displacements
  - Average is 2.5 bytes

i486 Pipeline

- Fetch
  - Load 16-bytes of instruction into prefetch buffer
- Decode1
  - Determine instruction length, instruction type
- Decode2
  - Compute memory address
  - Generate immediate operands
- Execute
  - Register Read
  - ALU operation
  - Memory read/write
- Write-Back
  - Update register file

Pipeline Stage Details

- Fetch
  - Moves 16 bytes of instruction stream into code queue
  - Not required every time
    - About 5 instructions fetched at once
    - Only useful if don't branch
  - Avoids need for separate instruction cache
- D1
  - Determine total instruction length
    - Signals code queue aligner where next instruction begins
  - May require two cycles
    - When multiple operands must be decoded
    - About 6% of "typical" DOS program

Stage Details (Cont.)

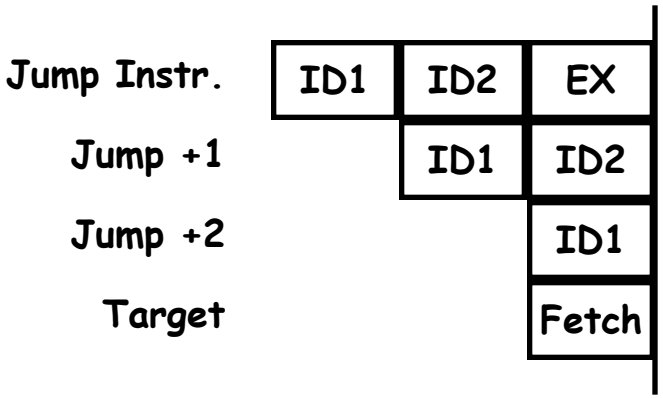
- D2
  - Extract memory displacements and immediate operands
  - Compute memory addresses
    - Add base register, and possibly scaled index register
  - May require two cycles
    - If index register involved, or both address & immediate operand
    - Approx. 5% of executed instructions
- EX
  - Read register operands
  - Compute ALU function
  - Read or write memory (data cache)
- WB
  - Update register result

Data Hazards

- Data Hazards

Generated	Used	Handling
ALU	**ALU**	**EX–EX Forwarding**
Load	**ALU**	**EX–EX Forwarding**
ALU	**Store**	**EX–EX Forwarding**
ALU	**Eff. Address**	**(Stall) + EX–ID2 Forwarding**

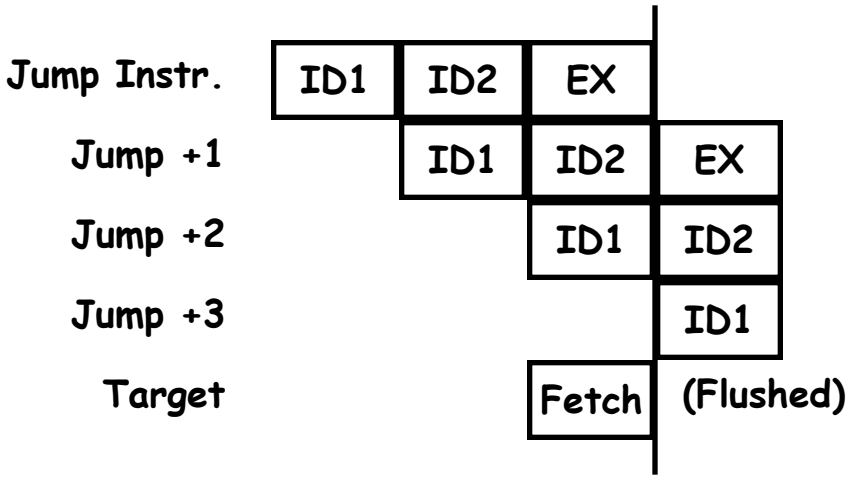
Control Hazards



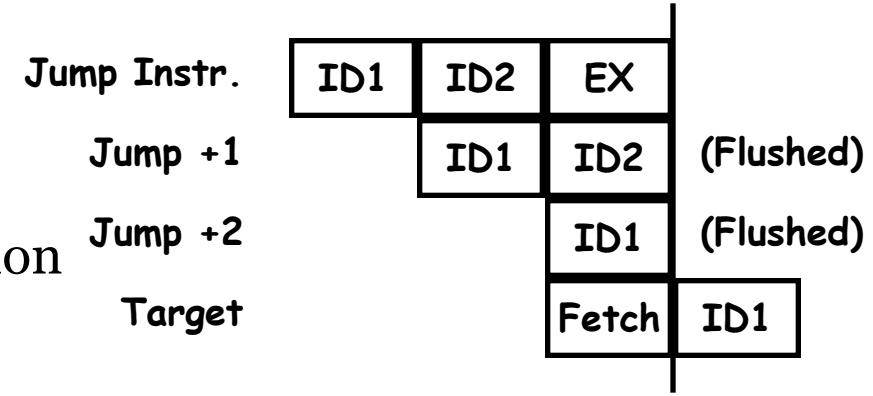
- Jump Instruction Processsing
  - Continue pipeline assuming branch not taken
  - Resolve branch condition in EX stage
  - Also speculatively fetch at target during EX stage

Control Hazards (Cont.)

- **Branch Not Taken**
  - Allow pipeline to continue.
  - Total of 1 cycle for instruction



- **Branch taken**
  - Flush instructions in pipe
  - Begin ID1 at target.
  - Total of 3 cycles for instruction



Comparison to 386

- Cycles Per Instruction

Instruction Type	386 Cycles	486 Cycles
Load	4	1
Store	2	1
ALU	2	1
Jump taken	9	3
Jump not taken	3	1
Call	9	3

- Reasons for Improvement
  - On chip cache
    - Faster loads & stores
  - More pipelining