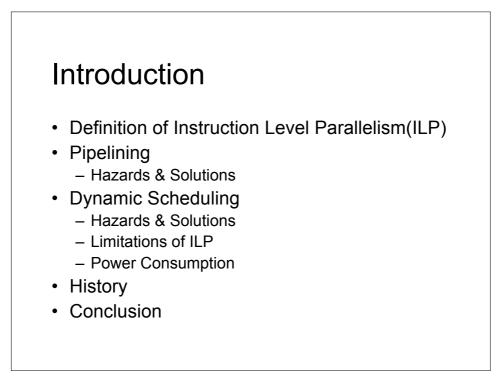
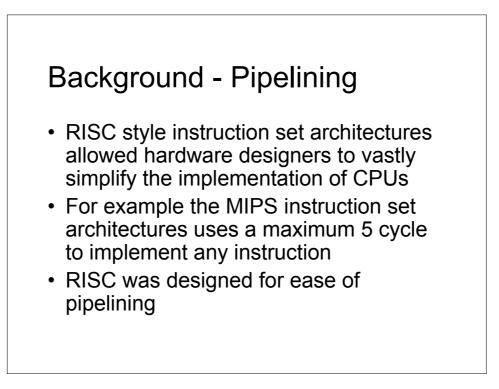
Instruction Level Parallelism

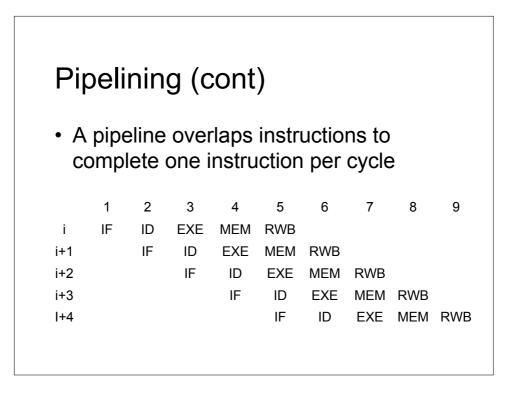
Software View of Computer Architecture COMP9244 Godfrey van der Linden 2006-04-06

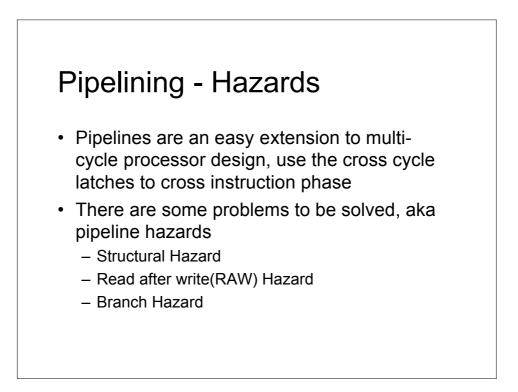




- Early processors would use more than one cycle to execute an instruction. Instruction per cycle (IPC) < 1
- Pipelines overlap instruction execution to achieve IPC = 1
- ILP can be defined as IPC > 1

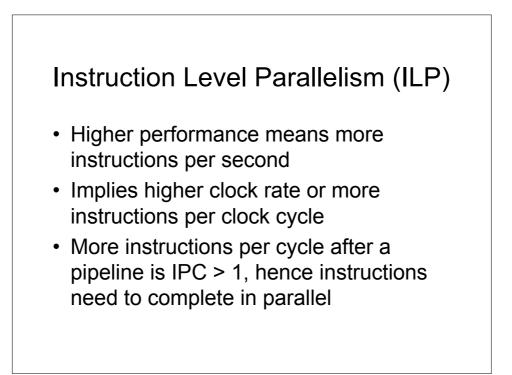


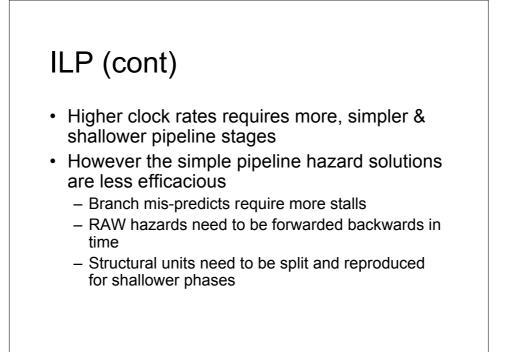


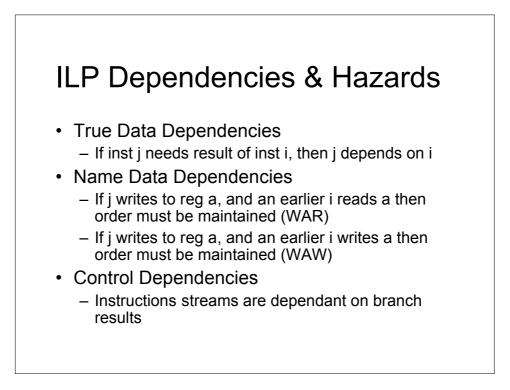


Pipelining - Hazard Solutions

- Stall pipeline until hazard clears
- RAW Hazard forwarding from one output phase direct to input phase
- Structural Hazard duplicate units, eg. Split data & instruction caches
- Branch Hazard predict a branch not taken and no-op & fetch if wrong

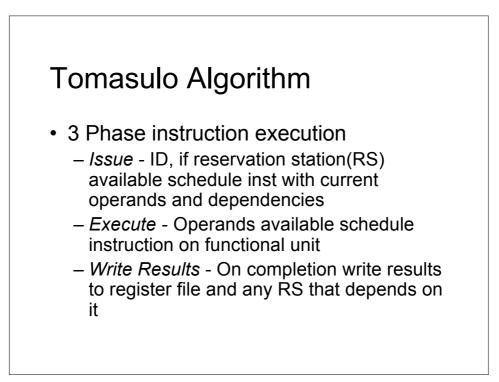








- Technique to execute instructions as soon as dependencies are satisfied
- Scoreboarding tracks instructions on a functional unit scheduling execution when data dependencies are satisfied
- Tomasulo extends this with dynamic renaming of registers to clear name dependencies

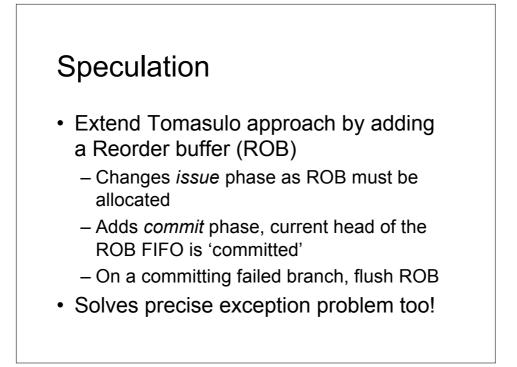


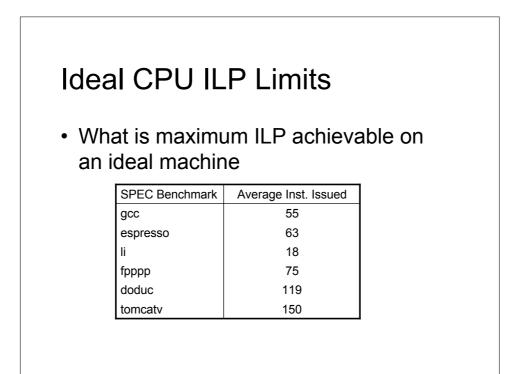


- Given a Tomasulo architecture can now add functional units to achieve even more parallelism
- Power4 has 2 Load/Store, 2 Fixed Point, 2 Floating Point, Branch and a CR unit



- 1 in 3 7 inst's is a branch. Control dependency stalls destroy IPC
- Modern unit tournament between local and global branch predictors
- · Stack of targets and returned addresses
- Branch prediction, today's typical unit achieves 95% accuracy
- OO virtual member functions present real problems to branch predictors without value prediction





SPEC/ Wdw	Inf.	256	128	64	32	16	8	4
gcc	10	10	10	9	8	6	4	3
espresso	15	15	13	10	8	6	4	2
li	12	12	11	11	9	6	4	3
fpppp	52	47	35	22	14	8	5	3
doduc	17	16	15	12	9	7	4	3
tomcatv	56	45	14	22	14	9	6	3

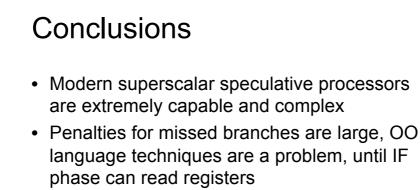
adiaahla CDUUU DUimita

Power Consumption

- [Li03] found IPC/Power were correlated for OS routines on a superscalar
- Using a cycle accurate power simulator of MIPS R10000, 50% of power used by data-path & pipeline structure
- The cite 3 other papers that collaborating their findings

History of ILP

- '59 IBM 7030 "stretch" Pipelining
- '64 CDC 6600 dynamic scheduling using scoreboarding
- '67 IBM 360/91 for dynamic scheduling Tomasulo
- '94/'95 1st gen superscalars: Pentium, AMD K6, MIPS 12000, PowerPC620
- End '90s 2nd gen: PIII, Athlon, Power4, Alpha 21264



• Other penalties are also significant, static scheduling for a processor architecture of instruction stream is recommended



- Statically scheduling code for processors is very difficult, use compiler technology from chip manufacturer
- Software engineers in high level languages can not influence code scheduling, except with branch hints
- With profiling help an engineer can achieve some 5x performance improvement over generated code in statically scheduled assembler [Grey05]