#### **Computer Architecture**

#### **Instruction Level Parallelism**

## Outline

- Instruction Level Parallelism (2.1)
- Compiler techniques for Exposing ILP (2.2)
- Reducing Branch Costs with Prediction (2.3)
- Overcoming Data Hazards with Dynamic Scheduling (2.4)
- Dynamic Scheduling: Examples and the Algorithm (2.5)
- Hardware-Based Speculation (2.6)
- Exploiting ILP using Multiple Issue and Static Scheduling (2.7)
- Exploiting ILP using Dynamic Scheduling, Multiple Issue, and Speculation (2.8)

## **Speculation to greater ILP**

- Greater ILP: Overcome control dependence by hardware speculating on outcome of branches and executing program as if guesses were correct
  - Speculation ⇒ fetch, issue, and execute instructions as if branch predictions were always correct
  - Dynamic scheduling ⇒ only fetches and issues instructions
- Essentially a data flow execution model: Operations execute as soon as their operands are available

# **Speculation to greater ILP**

- 3 components of HW-based speculation:
- 1. Dynamic branch prediction to choose which instructions to execute
- 2. Speculation to allow execution of instructions before control dependences are resolved
   + ability to undo effects of incorrectly speculated sequence
- 3. Dynamic scheduling to deal with scheduling of different combinations of basic blocks

## **Adding Speculation to Tomasulo**

- Must separate execution from allowing instruction to finish or "commit"
- This additional step called instruction commit
- When an instruction is no longer speculative, allow it to update the register file or memory
- Requires additional set of buffers to hold results of instructions that have finished execution but have not committed
- This reorder buffer (ROB) is also used to pass results among instructions that may be speculated

# **Reorder Buffer (ROB)**

- In Tomasulo's algorithm, once an instruction writes its result, any subsequently issued instructions will find result in the register file
- With speculation, the register file is not updated until the instruction commits
  - (we know definitively that the instruction should execute)
- Thus, the ROB supplies operands in interval between completion of instruction execution and instruction commit
  - ROB is a source of operands for instructions, just as reservation stations (RS) provide operands in Tomasulo's algorithm
  - ROB extends architectured registers like RS

## **Reorder Buffer Entry**

- Each entry in the ROB contains four fields:
- 1. Instruction type
  - a branch (has no destination result), a store (has a memory address destination), or a register operation (ALU operation or load, which has register destinations)
- 2. Destination
  - Register number (for loads and ALU operations) or memory address (for stores) where the instruction result should be written
- 3. Value
  - Value of instruction result until the instruction commits
- 4. Ready
  - Indicates that instruction has completed execution, and the value is ready

## **Reorder Buffer operation**

- Holds instructions in FIFO order, exactly as issued
- When instructions complete, results placed into ROB
  - Supplies operands to other instruction between execution complete & commit ⇒ more registers like RS
  - Tag results with ROB buffer number instead of reservation station
- Instructions commit ⇒values at head of ROB placed in registers



### Recall: 4 Steps of Speculative Tomasulo Algorithm

#### **1. Issue**—get instruction from FP Op Queue

If reservation station and reorder buffer slot free, issue instr & send operands & reorder buffer no. for destination (this stage sometimes called "dispatch")

#### **2. Execution**—operate on operands (EX)

When both operands ready then execute; if not ready, watch CDB for result; when both in reservation station, execute; checks RAW (sometimes called "issue")

#### **3. Write result**—finish execution (WB)

Write on Common Data Bus to all awaiting FUs & reorder buffer; mark reservation station available.

#### 4. Commit—update register with reorder result

When instr. at head of reorder buffer & result present, update register with result (or store to memory) and remove instr from reorder buffer. Mispredicted branch flushes reorder buffer (sometimes called "graduation")

| Status                         | Wait until  | Action or bookkeeping   |  |  |  |
|--------------------------------|---|---|--|--|--|
| Issue<br>all<br>instructions   | $\begin{array}{ccc} & \mbox{if (RegisterStat[rs].Busy)/*in-flight instr. writes rs*/}\\ & \mbox{(h }\leftarrow \mbox{ RegisterStat[rs].Reorder;}\\ & \mbox{if (ROB[h].Ready)/* Instr completed already */}\\ & \mbox{(RS[r].Vj }\leftarrow \mbox{ ROB[h].Value; RS[r].0j }\leftarrow \mbox{0;)}\\ & \mbox{else (RS[r].0j }\leftarrow \mbox{h;} /* \mbox{ wait for instruction */}\\ & \mbox{Reservation}\\ & \mbox{station (r)}\\ & \mbox{and} \\ & \mbox{ROB[b].Instruction }\leftarrow \mbox{opcode; ROB[b].Dest }\leftarrow \mbox{rd;ROB[b].Reservation} \\ & \mbox{ROB[b].Reservation} \\ & \mbox{ROB[b].Instruction }\leftarrow \mbox{opcode; ROB[b].Dest }\leftarrow \mbox{rd;ROB[b].Reservation} \\ & \mbox{ROB[b].Reservation} \\ & ROB[b].Re$ |   |  |  |  |
| FP<br>operations<br>and stores | ROB (6)<br>both available   | if (RegisterStat[rt].Busy) /*in-flight instr writes rt*/<br>{h $\leftarrow$ RegisterStat[rt].Reorder;<br>if (ROB[h].Ready)/* Instr completed already */<br>{RS[r].Vk $\leftarrow$ ROB[h].Value; RS[r].Qk $\leftarrow$ 0;}<br>else {RS[r].Qk $\leftarrow$ h;} /* wait for instruction */<br>} else {RS[r].Vk $\leftarrow$ Regs[rt]; RS[r].Qk $\leftarrow$ 0;}; |  |  |  |
| FP<br>operations               |   | RegisterStat[rd].Reorder $\leftarrow$ b; RegisterStat[rd].Busy $\leftarrow$ yes; ROB[b].Dest $\leftarrow$ rd;   |  |  |  |
| Loads                          |   | $RS[r].A \leftarrow imm; RegisterStat[rt].Reorder \leftarrow b; RegisterStat[rt].Busy \leftarrow yes; ROB[b].Dest \leftarrow rt;$   |  |  |  |
| Stores                         |   | $RS[r].A \leftarrow imm;$   |  |  |  |
| Execute<br>FP op               | (RS[r].Qj == 0) and $(RS[r].Qk == 0)$   | Compute results-operands are in Vj and Vk   |  |  |  |
| Load step 1                    | (RS[r].Qj == 0) and<br>there are no stores<br>earlier in the queue  | $RS[r].A \leftarrow RS[r].Vj + RS[r].A;$  |  |  |  |
| Load step 2                    | Load step 1 done<br>and all stores earlier<br>in ROB have<br>different address  | Read from Mem[RS[r].A]  |  |  |  |
| Store                          | (RS[r].Qj == 0) and<br>store at queue head  | <pre>ROB[h].Address ← RS[r].Vj + RS[r].A;</pre>   |  |  |  |
| Write result<br>all but store  | Execution done at r<br>and CDB available  | $\begin{array}{l} b \leftarrow RS[r].Dest; RS[r].Busy \leftarrow no; \\ \forall x (if (RS[x].Qj==b) (RS[x].Vj \leftarrow result; RS[x].Qj \leftarrow 0)); \\ \forall x (if (RS[x].Qk==b) (RS[x].Vk \leftarrow result; RS[x].Qk \leftarrow 0)); \\ ROB[b].Value \leftarrow result; ROB[b].Ready \leftarrow yes; \end{array}$                                   |  |  |  |
| Store                          | Execution done at r<br>and (RS[r].Qk ==<br>0)   | ROB[h].Value ← RS[r].Vk;  |  |  |  |
| Commit                         | Instruction is at the<br>head of the ROB<br>(entry h) and<br>ROB[h].ready ==<br>yes   | <pre>d ← ROB[h].Dest; /* register dest, if exists */ if (ROB[h].Instruction==Branch)     {if (branch is mispredicted)     {clear ROB[h]. RegisterStat; fetch branch dest;};} else if (ROB[h].Instruction==Store)</pre>  |  |  |  |

2007/4/25

















# Implication (Example in Text Book)

- The processor with the ROB can dynamically execute code while maintaining a precise interrupt model.
  - For example, if the MUL.D instruction caused an interrupt, we could simply wait until it reached the head of the ROB and take the interrupt, flushing any other pending instructions from the ROB.
     Because instruction commit happens in order, this yields a precise exception.
  - In the example using Tomasulo's algorithm, the SUB.D and ADD.D instructions could both complete before the MUL.D raised the exception.

## **Avoiding Memory Hazards**

- WAW and WAR hazards through memory are eliminated with speculation because actual updating of memory occurs in order, when a store is at head of the ROB, and hence, no earlier loads or stores can still be pending
- RAW hazards through memory are maintained by two restrictions:
  - 1. not allowing a load to initiate the second step of its execution if any active ROB entry occupied by a store has a Destination field that matches the value of the A field of the load, and
  - 2. maintaining the program order for the computation of an effective address of a load with respect to all earlier stores.
- these restrictions ensure that any load that accesses a memory location written to by an earlier store cannot perform the memory access until the store has written the data

## Outline

- Instruction Level Parallelism (2.1)
- Compiler techniques for Exposing ILP (2.2)
- Reducing Branch Costs with Prediction (2.3)
- Overcoming Data Hazards with Dynamic Scheduling (2.4)
- Dynamic Scheduling: Examples and the Algorithm (2.5)
- Hardware-Based Speculation (2.6)
- Exploiting ILP using Multiple Issue and Static Scheduling (2.7)
- Exploiting ILP using Dynamic Scheduling, Multiple Issue, and Speculation (2.8)

# **Getting CPI below 1**

- $CPI \ge 1$  if issue only 1 instruction every clock cycle
  - The goal of the multiple-issue processors is to allow multiple instructions to issue in a clock cycle.
- Multiple-issue processors come in 3 flavors:
  - 1. statically-scheduled superscalar processors,
  - 2. dynamically-scheduled superscalar processors, and
  - 3. VLIW (very long instruction word) processors

## **Multiple-Issue Processors**

- 2 types of superscalar processors issue varying numbers of instructions per clock
  - use in-order execution if they are statically scheduled, or
  - out-of-order execution if they are dynamically scheduled
- VLIW processors, in contrast, issue a fixed number of instructions formatted either as one large instruction or as a fixed instruction packet with the parallelism among instructions explicitly indicated by the instruction (Intel/HP Itanium –IA-63)

## **VLIW: Very Large Instruction Word**

- Each "instruction" has explicit coding for multiple operations
  - In IA-64, grouping called a "packet"
  - In Transmeta, grouping called a "molecule" (with "atoms" as ops)

#### Tradeoff instruction space for simple decoding

- The long instruction word has room for many operations
- By definition, all the operations the compiler puts in the long instruction word are independent => execute in parallel
- E.g., 2 integer operations, 2 FP ops, 2 Memory refs, 1 branch
  - » 16 to 24 bits per field => 7\*16 or 112 bits to 7\*24 or 168 bits wide
- Need compiling technique that schedules across several branches

#### **Recall: Unrolled Loop that Minimizes Stalls for Scalar**

| 1 Loop: | L.D    | F0,0(R1)    |   | L.D to A   | D  |
|---------|--------|-------------|---|------------|----|
| 2       | L.D    | F6,-8(R1)   |   | ADD.D      | to |
| 3       | L.D    | F10,-16(R1) |   |            |    |
| 4       | L.D    | F14,-24(R1) |   |            |    |
| 5       | ADD.D  | F4,F0,F2    |   |            |    |
| 6       | ADD.D  | F8,F6,F2    |   |            |    |
| 7       | ADD.D  | F12,F10,F2  |   |            |    |
| 8       | ADD.D  | F16,F14,F2  |   |            |    |
| 9       | S.D    | 0(R1),F4    |   |            |    |
| 10      | S.D    | -8(R1),F8   |   |            |    |
| 11      | S.D    | -16(R1),F12 |   |            |    |
| 12      | DSUBUI | R1,R1,#32   |   |            |    |
| 13      | BNEZ   | R1,LOOP     |   |            |    |
| 14      | S.D    | 8(R1),F16   | ; | 8-32 = -24 |    |

L.D to ADD.D: 1 Cycle ADD.D to S.D: 2 Cycles

#### 14 clock cycles, or 3.5 per iteration

# **Loop Unrolling in VLIW**

| Memory<br>reference 1 | Memory<br>reference 2 | FP operation 1   | FP operation 2   | Integer<br>operation/branch |
|-----------------------|-----------------------|------------------|------------------|-----------------------------|
| L.D F0,0(R1)          | L.D F6,-8(R1)         |                  |                  |                             |
| L.D F10,-16(R1)       | L.D F14,-24(R1)       |                  |                  |                             |
| L.D F18,-32(R1)       | L.D F22,-40(R1)       | ADD.D F4,F0,F2   | ADD.D F8,F6,F2   |                             |
| L.D F26,-48(R1)       |                       | ADD.D F12,F10,F2 | ADD.D F16,F14,F2 |                             |
|                       |                       | ADD.D F20,F18,F2 | ADD.D F24,F22,F2 |                             |
| S.D F4,0(R1)          | S.D F8,-8(R1)         | ADD.D F28,F26,F2 |                  |                             |
| S.D F12,-16(R1)       | S.D F16,-24(R1)       |                  |                  | DADDUI R1,R1,#-56           |
| S.D F20,24(R1)        | S.D F24,16(R1)        |                  |                  |                             |
| S.D F28,8(R1)         |                       |                  |                  | BNE R1, R2, Loop            |

- Unrolled 7 times to avoid delays
- 7 results in 9 clocks, or 1.3 clocks per iteration (1.8X)
- Average: 2.5 ops per clock, 50% efficiency
- Note: Need more registers in VLIW (15)

## **Problems with 1st Generation VLIW**

#### Increase in code size

- generating enough operations in a straight-line code fragment requires ambitiously unrolling loops
- whenever VLIW instructions are not full, unused functional units translate to wasted bits in instruction encoding

To combat this code size increase, clever encodings are sometimes used.

Another technique is to compress the instructions in main memory and expand them when they are read into the cache or are decoded.

## **Problems with 1st Generation VLIW**

- Operated in lock-step; no hazard detection HW
  - A stall in any functional unit pipeline caused entire processor to stall, since all functional units must be kept synchronized
  - Compiler might prediction function units, but caches hard to predict
- Binary code compatibility
  - Pure VLIW 

     different numbers of functional units and unit latencies require different versions of the code

## Intel/HP IA-64 "Explicitly Parallel Instruction Computer (EPIC)"

- <u>IA-64</u>: instruction set architecture
- 128 64-bit integer registers + 128 82-bit floating point registers
- Hardware checks dependencies (interlocks => binary compatibility over time)
- Extension for more aggressive software speculation
- Preserving binary compatibility
- Predicated execution (select 1 out of 64 1-bit flags)
   => 40% fewer mispredictions?
- **<u>Itanium</u><sup>™</sup>** was first implementation (2001)
  - Highly parallel and deeply pipelined hardware at 800Mhz
  - 6-wide, 10-stage pipeline at 800Mhz on 0.18  $\mu$  process
- **Itanium 2<sup>™</sup> is name of 2nd implementation (2005)** 
  - 6-wide, 8-stage pipeline at 1666Mhz on 0.13  $\mu$  process
  - Caches: 32 KB I, 32 KB D, 128 KB L2I, 128 KB L2D, 9216 KB L3

## Outline

- Instruction Level Parallelism (2.1)
- Compiler techniques for Exposing ILP (2.2)
- Reducing Branch Costs with Prediction (2.3)
- Overcoming Data Hazards with Dynamic Scheduling (2.4)
- Dynamic Scheduling: Examples and the Algorithm (2.5)
- Hardware-Based Speculation (2.6)
- Exploiting ILP using Multiple Issue and Static Scheduling (2.7)
- Exploiting ILP using Dynamic Scheduling, Multiple Issue, and Speculation (2.8)

# **Put All Together**

- To gain the full advantage of dynamic scheduling we will allow the pipeline to issue any combinations in a clock, using the scheduling hardware to actually assign operations to the integer and floating-point unit.
- Speculation can be advantageous when there are data-dependent branches, which otherwise would limit the performance.

# Put All Together (cont.)

 Consider the execution of the following loop, which increments each element of an integer array, on a two issue processor, once without speculation and once with speculation:

| Loop : | LD     | R2,0(R1)   | ;R2=array element           |
|--------|--------|------------|-----------------------------|
|        | DADDIU | R2,R2,#1   | ;increment R2               |
|        | SD     | R2,0(R1)   | ;store result               |
|        | DADDIU | R1,R1,#8   | ;increment pointer          |
|        | BNE    | R2,R3,Loop | ;branch if not last element |

- Assumptions:
  - Separate integer functional units for effective address calculation, for ALU operations, and for branch condition evaluation.
  - Up to two instructions of any type can commit per clock.

#### **Two-Issue Dynamically Scheduled Processor without Speculation (1)**

| Iteration<br>number | Instructions |            | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment     |
|---------------------|--------------|------------|---------------------------------------|---|--|--|-------------|
| 1                   | LD           | R2,0(R1)   | 1                                     |   |  |  | First issue |
| 1                   | DADDIU       | R2,R2,#1   | 1                                     |   |  |  |             |
| 1                   | SD           | R2,0(R1)   |                                       |   |  |  |             |
| 1                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |             |
| 1                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |
| 2                   | LD           | R2,0(R1)   |                                       |   |  |  |             |
| 2                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |             |
| 2                   | SD           | R2,0(R1)   |                                       |   |  |  |             |
| 2                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |             |
| 2                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |
| 3                   | LD           | R2,0(R1)   |                                       |   |  |  |             |
| 3                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |             |
| 3                   | SD           | R2,0(R1)   |                                       |   |  |  |             |
| 3                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |             |
| 3                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |

#### **Two-Issue Dynamically Scheduled Processor without Speculation (2)**

| lteration<br>number | Instructions |            | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment     |
|---------------------|--------------|------------|---------------------------------------|---|--|--|-------------|
| 1                   | LD           | R2,0(R1)   | 1                                     | 2                                       |  |  | First issue |
| 1                   | DADDIU       | R2,R2,#1   | 1                                     |   |  |  | Wait for LD |
| 1                   | SD           | R2,0(R1)   | 2                                     |   |  |  |             |
| 1                   | DADDIU       | R1,R1,#8   | 2                                     |   |  |  |             |
| 1                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |
| 2                   | LD           | R2,0(R1)   |                                       |   |  |  |             |
| 2                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |             |
| 2                   | SD           | R2,0(R1)   |                                       |   |  |  |             |
| 2                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |             |
| 2                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |
| 3                   | LD           | R2,0(R1)   |                                       |   |  |  |             |
| 3                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |             |
| 3                   | SD           | R2,0(R1)   |                                       |   |  |  |             |
| 3                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |             |
| 3                   | BNE          | R2,R3,LOOP |                                       |   |  |  |             |

#### **Two-Issue Dynamically Scheduled Processor without Speculation (3)**

| Iteration<br>number | Instructions |            | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD           | R2,0(R1)   | 1                                     | 2                                       | 3  |  | First issue      |
| 1                   | DADDIU       | R2,R2,#1   | 1                                     |   |  |  | Wait for LD      |
| 1                   | SD           | R2,0(R1)   | 2                                     | 3                                       |  |  |                  |
| 1                   | DADDIU       | R1,R1,#8   | 2                                     | 3                                       |  |  | Execute directly |
| 1                   | BNE          | R2,R3,LOOP | 3                                     |   |  |  |                  |
| 2                   | LD           | R2,0(R1)   |                                       |   |  |  |                  |
| 2                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |                  |
| 2                   | SD           | R2,0(R1)   |                                       |   |  |  |                  |
| 2                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |                  |
| 2                   | BNE          | R2,R3,LOOP |                                       |   |  |  |                  |
| 3                   | LD           | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |                  |
| 3                   | SD           | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |                  |
| 3                   | BNE          | R2,R3,LOOP |                                       |   |  |  |                  |

#### **Two-Issue Dynamically Scheduled Processor without Speculation (4)**

| lteration<br>number | Instructions |            | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD           | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU       | R2,R2,#1   | 1                                     |   |  |  | Wait for LD      |
| 1                   | SD           | R2,0(R1)   | 2                                     | 3                                       |  |  | Wait for DADDIU  |
| 1                   | DADDIU       | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE          | R2,R3,LOOP | 3                                     |   |  |  | Wait for DADDIU  |
| 2                   | LD           | R2,0(R1)   | 4                                     |   |  |  |                  |
| 2                   | DADDIU       | R2,R2,#1   | 4                                     |   |  |  |                  |
| 2                   | SD           | R2,0(R1)   |                                       |   |  |  |                  |
| 2                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |                  |
| 2                   | BNE          | R2,R3,LOOP |                                       |   |  |  |                  |
| 3                   | LD           | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU       | R2,R2,#1   |                                       |   |  |  |                  |
| 3                   | SD           | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU       | R1,R1,#8   |                                       |   |  |  |                  |
| 3                   | BNE          | R2,R3,LOOP |                                       |   |  |  |                  |
# **Two-Issue Dynamically Scheduled Processor without Speculation (5)**

| Iteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  |  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       |  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     |   |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     |   |  |  | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     |   |  |  |                  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     |   |  |  |                  |
| 2                   | BNE    | R2,R3,LOOP |                                       |   |  |  |                  |
| 3                   | LD     | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU | R2,R2,#1   |                                       |   |  |  |                  |
| 3                   | SD     | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU | R1,R1,#8   |                                       |   |  |  |                  |
| 3                   | BNE    | R2,R3,LOOP |                                       |   |  |  |                  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (6)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       |  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     |   |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     |   |  |  | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     |   |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     |   |  |  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  |                  |
| 3                   | LD     | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU | R2,R2,#1   |                                       |   |  |  |                  |
| 3                   | SD     | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU | R1,R1,#8   |                                       |   |  |  |                  |
| 3                   | BNE    | R2,R3,LOOP |                                       |   |  |  |                  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (7)**

| Iteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     |   |  |  | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     |   |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     |   |  |  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  |                  |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  |                  |
| 3                   | SD     | R2,0(R1)   |                                       |   |  |  |                  |
| 3                   | DADDIU | R1,R1,#8   |                                       |   |  |  |                  |
| 3                   | BNE    | R2,R3,LOOP |                                       |   |  |  |                  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (8)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       |  |  | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     |   |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  |  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  |                  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  |                  |
| 3                   | BNE    | R2,R3,LOOP |                                       |   |  |  |                  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (9)**

| Iteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  |  | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  |                  |

# Two-Issue Dynamically Scheduled Processor without Speculation (10)

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     |   |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (11)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  |  | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# Two-Issue Dynamically Scheduled Processor without Speculation (12)

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       |  |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     |   |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# Two-Issue Dynamically Scheduled Processor without Speculation (13)

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     |   |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     |   |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (14)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      |  |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     |   |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  |  | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# Two-Issue Dynamically Scheduled Processor without Speculation (15)

| Iteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      | 15   |  | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     | 15                                      |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  | 15                                       | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (16)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      | 15   | 16                                       | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     |   |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     | 15                                      |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  | 15                                       | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (17)**

| Iteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      | 15   | 16                                       | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     | 17                                      |  |  | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     | 15                                      |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  | 15                                       | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (18)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      | 15   | 16                                       | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     | 17                                      |  | 18                                       | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     | 15                                      |  |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  | 15                                       | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     |   |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor without Speculation (19)**

| lteration<br>number | Ins    | structions | lssues at<br>clock<br>cycle<br>number | Executes<br>at clock<br>cycle<br>number | Memory<br>access at<br>clock cycle<br>number | Write CDB<br>at clock<br>cycle<br>number | Comment          |
|---------------------|--------|------------|---------------------------------------|---|--|--|------------------|
| 1                   | LD     | R2,0(R1)   | 1                                     | 2                                       | 3  | 4  | First issue      |
| 1                   | DADDIU | R2,R2,#1   | 1                                     | 5                                       |  | 6  | Wait for LD      |
| 1                   | SD     | R2,0(R1)   | 2                                     | 3                                       | 7  |  | Wait for DADDIU  |
| 1                   | DADDIU | R1,R1,#8   | 2                                     | 3                                       |  | 4  | Execute directly |
| 1                   | BNE    | R2,R3,LOOP | 3                                     | 7                                       |  |  | Wait for DADDIU  |
| 2                   | LD     | R2,0(R1)   | 4                                     | 8                                       | 9  | 10                                       | Wait for BNE     |
| 2                   | DADDIU | R2,R2,#1   | 4                                     | 11                                      |  | 12                                       | Wait for LD      |
| 2                   | SD     | R2,0(R1)   | 5                                     | 9                                       | 13   |  | Wait for DADDIU  |
| 2                   | DADDIU | R1,R1,#8   | 5                                     | 8                                       |  | 9  | Wait for BNE     |
| 2                   | BNE    | R2,R3,LOOP | 6                                     | 13                                      |  |  | Wait for DADDIU  |
| 3                   | LD     | R2,0(R1)   | 7                                     | 14                                      | 15   | 16                                       | Wait for BNE     |
| 3                   | DADDIU | R2,R2,#1   | 7                                     | 17                                      |  | 18                                       | Wait for LD      |
| 3                   | SD     | R2,0(R1)   | 8                                     | 15                                      | 19   |  | Wait for DADDIU  |
| 3                   | DADDIU | R1,R1,#8   | 8                                     | 14                                      |  | 15                                       | Wait for BNE     |
| 3                   | BNE    | R2,R3,LOOP | 9                                     | 19                                      |  |  | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor with Speculation (1)**

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment     |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------|
| 1 | LD R2,0(R1)      | 1                            |                                |                                      |                                    |                               | First issue |
| 1 | DADDIU R2,R2,#1  | 1                            |                                |                                      |                                    |                               |             |
| 1 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 1 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 1 | BNE R2, R3, LOOP |                              |                                |                                      |                                    |                               |             |
| 2 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 2 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 2 | BNE R2,R3,LOOP   |                              |                                |                                      |                                    |                               |             |
| 3 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 3 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 3 | BNE R2,R3,LOOP   |                              |                                |                                      |                                    |                               |             |

# Two-Issue Dynamically Scheduled Processor with Speculation (2)

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment     |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              |                                      |                                    |                               | First issue |
| 1 | DADDIU R2,R2,#1  | 1                            |                                |                                      |                                    |                               | Wait for LD |
| 1 | SD R2,0(R1)      | 2                            |                                |                                      |                                    |                               |             |
| 1 | DADDIU R1,R1,#8  | 2                            |                                |                                      |                                    |                               |             |
| 1 | BNE R2, R3, LOOP |                              |                                |                                      |                                    |                               |             |
| 2 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 2 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 2 | BNE R2, R3, LOOP |                              |                                |                                      |                                    |                               |             |
| 3 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 3 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 3 | BNE R2,R3,LOOP   |                              |                                |                                      |                                    |                               |             |

# **Two-Issue Dynamically Scheduled Processor with Speculation (3)**

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment     |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    |                                    |                               | First issue |
| 1 | DADDIU R2,R2,#1  | 1                            |                                |                                      |                                    |                               | Wait for LD |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    |                               |             |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      |                                    |                               |             |
| 1 | BNE R2, R3, LOOP | 3                            |                                |                                      |                                    |                               |             |
| 2 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 2 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 2 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 2 | BNE R2, R3, LOOP |                              |                                |                                      |                                    |                               |             |
| 3 | LD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R2,R2,#1  |                              |                                |                                      |                                    |                               |             |
| 3 | SD R2,0(R1)      |                              |                                |                                      |                                    |                               |             |
| 3 | DADDIU R1,R1,#8  |                              |                                |                                      |                                    |                               |             |
| 3 | BNE R2, R3, LOOP |                              |                                |                                      |                                    |                               |             |

# **Two-Issue Dynamically Scheduled Processor with Speculation (4)**

|   | Instructions    | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment         |
|---|-----------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-----------------|
| 1 | LD R2,0(R1)     | 1                            | 2                              | 3                                    | 4                                  |                               | First issue     |
| 1 | DADDIU R2,R2,#1 | 1                            |                                |                                      |                                    |                               | Wait for LD     |
| 1 | SD R2,0(R1)     | 2                            | 3                              |                                      |                                    |                               | Wait for DADDIU |
| 1 | DADDIU R1,R1,#8 | 2                            | 3                              |                                      | 4                                  |                               |                 |
| 1 | BNE R2,R3,LOOP  | 3                            |                                |                                      |                                    |                               | Wait for DADDIU |
| 2 | LD R2,0(R1)     | 4                            |                                |                                      |                                    |                               |                 |
| 2 | DADDIU R2,R2,#1 | 4                            |                                |                                      |                                    |                               |                 |
| 2 | SD R2,0(R1)     |                              |                                |                                      |                                    |                               |                 |
| 2 | DADDIU R1,R1,#8 |                              |                                |                                      |                                    |                               |                 |
| 2 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                 |
| 3 | LD R2,0(R1)     |                              |                                |                                      |                                    |                               |                 |
| 3 | DADDIU R2,R2,#1 |                              |                                |                                      |                                    |                               |                 |
| 3 | SD R2,0(R1)     |                              |                                |                                      |                                    |                               |                 |
| 3 | DADDIU R1,R1,#8 |                              |                                |                                      |                                    |                               |                 |
| 3 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                 |

# Two-Issue Dynamically Scheduled Processor with Speculation (5)

|   | Instructions    | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|-----------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)     | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1 | 1                            | 5                              |                                      |                                    |                               | Wait for LD      |
| 1 | SD R2,0(R1)     | 2                            | 3                              |                                      |                                    |                               | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8 | 2                            | 3                              |                                      | 4                                  |                               | Commit in order  |
| 1 | BNE R2,R3,LOOP  | 3                            |                                |                                      |                                    |                               | Wait for DADDIU  |
| 2 | LD R2,0(R1)     | 4                            | 5                              |                                      |                                    |                               | No execute delay |
| 2 | DADDIU R2,R2,#1 | 4                            |                                |                                      |                                    |                               | Wait for LD      |
| 2 | SD R2,0(R1)     | 5                            |                                |                                      |                                    |                               |                  |
| 2 | DADDIU R1,R1,#8 | 5                            |                                |                                      |                                    |                               |                  |
| 2 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                  |
| 3 | LD R2,0(R1)     |                              |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R2,R2,#1 |                              |                                |                                      |                                    |                               |                  |
| 3 | SD R2,0(R1)     |                              |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R1,R1,#8 |                              |                                |                                      |                                    |                               |                  |
| 3 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                  |

# Two-Issue Dynamically Scheduled Processor with Speculation (6)

|   | Instructions    | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|-----------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)     | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1 | 1                            | 5                              |                                      | 6                                  |                               | Wait for LD      |
| 1 | SD R2,0(R1)     | 2                            | 3                              |                                      |                                    |                               | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8 | 2                            | 3                              |                                      | 4                                  |                               | Commit in order  |
| 1 | BNE R2,R3,LOOP  | 3                            |                                |                                      |                                    |                               | Wait for DADDIU  |
| 2 | LD R2,0(R1)     | 4                            | 5                              | 6                                    |                                    |                               | No execute delay |
| 2 | DADDIU R2,R2,#1 | 4                            |                                |                                      |                                    |                               | Wait for LD      |
| 2 | SD R2,0(R1)     | 5                            | 6                              |                                      |                                    |                               |                  |
| 2 | DADDIU R1,R1,#8 | 5                            | 6                              |                                      |                                    |                               |                  |
| 2 | BNE R2,R3,LOOP  | 6                            |                                |                                      |                                    |                               |                  |
| 3 | LD R2,0(R1)     |                              |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R2,R2,#1 |                              |                                |                                      |                                    |                               |                  |
| 3 | SD R2,0(R1)     |                              |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R1,R1,#8 |                              |                                |                                      |                                    |                               |                  |
| 3 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                  |

# Two-Issue Dynamically Scheduled Processor with Speculation (7)

|   | Instructions    | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|-----------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)     | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1 | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD      |
| 1 | SD R2,0(R1)     | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8 | 2                            | 3                              |                                      | 4                                  |                               | Commit in order  |
| 1 | BNE R2,R3,LOOP  | 3                            | 7                              |                                      |                                    |                               | Wait for DADDIU  |
| 2 | LD R2,0(R1)     | 4                            | 5                              | 6                                    | 7                                  |                               | No execute delay |
| 2 | DADDIU R2,R2,#1 | 4                            |                                |                                      |                                    |                               | Wait for LD      |
| 2 | SD R2,0(R1)     | 5                            | 6                              |                                      |                                    |                               | Wait for DADDIU  |
| 2 | DADDIU R1,R1,#8 | 5                            | 6                              |                                      | 7                                  |                               |                  |
| 2 | BNE R2,R3,LOOP  | 6                            |                                |                                      |                                    |                               | Wait for DADDIU  |
| 3 | LD R2,0(R1)     | 7                            |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R2,R2,#1 | 7                            |                                |                                      |                                    |                               |                  |
| 3 | SD R2,0(R1)     |                              |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R1,R1,#8 |                              |                                |                                      |                                    |                               |                  |
| 3 | BNE R2,R3,LOOP  |                              |                                |                                      |                                    |                               |                  |

# Two-Issue Dynamically Scheduled Processor with Speculation (8)

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD      |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order  |
| 1 | BNE R2, R3, LOOP | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU  |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  |                               | No execute delay |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      |                                    |                               | Wait for LD      |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    |                               | Wait for DADDIU  |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  |                               | Commit in order  |
| 2 | BNE R2,R3,LOOP   | 6                            |                                |                                      |                                    |                               | Wait for DADDIU  |
| 3 | LD R2,0(R1)      | 7                            | 8                              |                                      |                                    |                               |                  |
| 3 | DADDIU R2,R2,#1  | 7                            |                                |                                      |                                    |                               |                  |
| 3 | SD R2,0(R1)      | 8                            |                                |                                      |                                    |                               |                  |
| 3 | DADDIU R1,R1,#8  | 8                            |                                |                                      |                                    |                               |                  |
| 3 | BNE R2,R3,LOOP   |                              |                                |                                      |                                    |                               |                  |

# **Two-Issue Dynamically Scheduled Processor with Speculation (9)**

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD      |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order  |
| 1 | BNE R2,R3,LOOP   | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU  |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      | 9                                  |                               | Wait for LD      |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    |                               | Wait for DADDIU  |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  |                               | Commit in order  |
| 2 | BNE R2, R3, LOOP | 6                            |                                |                                      |                                    |                               | Wait for DADDIU  |
| 3 | LD R2,0(R1)      | 7                            | 8                              | 9                                    |                                    |                               |                  |
| 3 | DADDIU R2,R2,#1  | 7                            |                                |                                      |                                    |                               | Wait for LD      |
| 3 | SD R2,0(R1)      | 8                            | 9                              |                                      |                                    |                               |                  |
| 3 | DADDIU R1,R1,#8  | 8                            | 9                              |                                      |                                    |                               | Execute earlier  |
| 3 | BNE R2,R3,LOOP   | 9                            |                                |                                      |                                    |                               |                  |

# Two-Issue Dynamically Scheduled Processor with Speculation (10)

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD      |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order  |
| 1 | BNE R2, R3, LOOP | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU  |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      | 9                                  | 10                            | Wait for LD      |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    | 10                            | Wait for DADDIU  |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  |                               | Commit in order  |
| 2 | BNE R2, R3, LOOP | 6                            | 10                             |                                      |                                    |                               | Wait for DADDIU  |
| 3 | LD R2,0(R1)      | 7                            | 8                              | 9                                    | 10                                 |                               |                  |
| 3 | DADDIU R2,R2,#1  | 7                            |                                |                                      |                                    |                               | Wait for LD      |
| 3 | SD R2,0(R1)      | 8                            | 9                              |                                      |                                    |                               | Wait for DADDIU  |
| 3 | DADDIU R1,R1,#8  | 8                            | 9                              |                                      | 10                                 |                               | Execute earlier  |
| 3 | BNE R2, R3, LOOP | 9                            |                                |                                      |                                    |                               | Wait for DADDIU  |

# **Two-Issue Dynamically Scheduled Processor with Speculation (11)**

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment          |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue      |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD      |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU  |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order  |
| 1 | BNE R2,R3,LOOP   | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU  |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      | 9                                  | 10                            | Wait for LD      |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    | 10                            | Wait for DADDIU  |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  | 11                            | Commit in order  |
| 2 | BNE R2,R3,LOOP   | 6                            | 10                             |                                      |                                    | 11                            | Wait for DADDIU  |
| 3 | LD R2,0(R1)      | 7                            | 8                              | 9                                    | 10                                 |                               |                  |
| 3 | DADDIU R2,R2,#1  | 7                            | 11                             |                                      |                                    |                               | Wait for LD      |
| 3 | SD R2,0(R1)      | 8                            | 9                              |                                      |                                    |                               | Wait for DADDIU  |
| 3 | DADDIU R1,R1,#8  | 8                            | 9                              |                                      | 10                                 |                               | Execute earlier  |
| 3 | BNE R2, R3, LOOP | 9                            |                                |                                      |                                    |                               | Wait for DADDIU  |

# Two-Issue Dynamically Scheduled Processor with Speculation (12)

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment           |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue       |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD       |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU   |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order   |
| 1 | BNE R2,R3,LOOP   | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU   |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay  |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      | 9                                  | 10                            | Wait for LD       |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    | 10                            | Wait for DADDIU   |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  | 11                            | Commit in order   |
| 2 | BNE R2,R3,LOOP   | 6                            | 10                             |                                      |                                    | 11                            | Wait for DADDIU   |
| 3 | LD R2,0(R1)      | 7                            | 8                              | 9                                    | 10                                 | 12                            | Earliest possible |
| 3 | DADDIU R2,R2,#1  | 7                            | 11                             |                                      | 12                                 |                               | Wait for LD       |
| 3 | SD R2,0(R1)      | 8                            | 9                              |                                      |                                    |                               | Wait for DADDIU   |
| 3 | DADDIU R1,R1,#8  | 8                            | 9                              |                                      | 10                                 |                               | Execute earlier   |
| 3 | BNE R2, R3, LOOP | 9                            |                                |                                      |                                    |                               | Wait for DADDIU   |

# Two-Issue Dynamically Scheduled Processor with Speculation (13)

|   | Instructions    | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment           |
|---|-----------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------------|
| 1 | LD R2,0(R1)     | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue       |
| 1 | DADDIU R2,R2,#1 | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD       |
| 1 | SD R2,0(R1)     | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU   |
| 1 | DADDIU R1,R1,#8 | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order   |
| 1 | BNE R2,R3,LOOP  | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU   |
| 2 | LD R2,0(R1)     | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay  |
| 2 | DADDIU R2,R2,#1 | 4                            | 8                              |                                      | 9                                  | 10                            | Wait for LD       |
| 2 | SD R2,0(R1)     | 5                            | 6                              |                                      |                                    | 10                            | Wait for DADDIU   |
| 2 | DADDIU R1,R1,#8 | 5                            | 6                              |                                      | 7                                  | 11                            | Commit in order   |
| 2 | BNE R2,R3,LOOP  | 6                            | 10                             |                                      |                                    | 11                            | Wait for DADDIU   |
| 3 | LD R2,0(R1)     | 7                            | 8                              | 9                                    | 10                                 | 12                            | Earliest possible |
| 3 | DADDIU R2,R2,#1 | 7                            | 11                             |                                      | 12                                 | 13                            | Wait for LD       |
| 3 | SD R2,0(R1)     | 8                            | 9                              |                                      |                                    | 13                            | Wait for DADDIU   |
| 3 | DADDIU R1,R1,#8 | 8                            | 9                              |                                      | 10                                 |                               | Execute earlier   |
| 3 | BNE R2,R3,LOOP  | 9                            | 13                             |                                      |                                    |                               | Wait for DADDIU   |

# Two-Issue Dynamically Scheduled Processor with Speculation (14)

|   | Instructions     | lssues<br>at clock<br>number | Executes<br>at clock<br>number | Read<br>access at<br>clock<br>number | Write<br>CDB at<br>clock<br>number | Commits<br>at clock<br>number | Comment           |
|---|------------------|------------------------------|--------------------------------|--------------------------------------|------------------------------------|-------------------------------|-------------------|
| 1 | LD R2,0(R1)      | 1                            | 2                              | 3                                    | 4                                  | 5                             | First issue       |
| 1 | DADDIU R2,R2,#1  | 1                            | 5                              |                                      | 6                                  | 7                             | Wait for LD       |
| 1 | SD R2,0(R1)      | 2                            | 3                              |                                      |                                    | 7                             | Wait for DADDIU   |
| 1 | DADDIU R1,R1,#8  | 2                            | 3                              |                                      | 4                                  | 8                             | Commit in order   |
| 1 | BNE R2,R3,LOOP   | 3                            | 7                              |                                      |                                    | 8                             | Wait for DADDIU   |
| 2 | LD R2,0(R1)      | 4                            | 5                              | 6                                    | 7                                  | 9                             | No execute delay  |
| 2 | DADDIU R2,R2,#1  | 4                            | 8                              |                                      | 9                                  | 10                            | Wait for LD       |
| 2 | SD R2,0(R1)      | 5                            | 6                              |                                      |                                    | 10                            | Wait for DADDIU   |
| 2 | DADDIU R1,R1,#8  | 5                            | 6                              |                                      | 7                                  | 11                            | Commit in order   |
| 2 | BNE R2, R3, LOOP | 6                            | 10                             |                                      |                                    | 11                            | Wait for DADDIU   |
| 3 | LD R2,0(R1)      | 7                            | 8                              | 9                                    | 10                                 | 12                            | Earliest possible |
| 3 | DADDIU R2,R2,#1  | 7                            | 11                             |                                      | 12                                 | 13                            | Wait for LD       |
| 3 | SD R2,0(R1)      | 8                            | 9                              |                                      |                                    | 13                            | Wait for DADDIU   |
| 3 | DADDIU R1,R1,#8  | 8                            | 9                              |                                      | 10                                 | 14                            | Execute earlier   |
| 3 | BNE R2, R3, LOOP | 9                            | 13                             |                                      |                                    | 14                            | Wait for DADDIU   |

# Outline

- Instruction Level Parallelism (2.1)
- Compiler techniques for Exposing ILP (2.2)
- Reducing Branch Costs with Prediction (2.3)
- Overcoming Data Hazards with Dynamic Scheduling (2.4)
- Dynamic Scheduling: Examples and the Algorithm (2.5)
- Hardware-Based Speculation (2.6)
- Exploiting ILP using Multiple Issue and Static Scheduling (2.7)
- Exploiting ILP using Dynamic Scheduling, Multiple Issue, and Speculation (2.8)
- Advanced Techniques for Instruction Delivery and Speculation (2.9)

# **Advanced Techniques**

- Increasing Instruction Fetch Bandwidth
  - Branch-Target Buffers
  - Return Address Predictors
  - Integrated Instruction Fetch Units
- Speculation: Implementation Issues and Extensions
  - Speculation Support: Register Renaming versus Reorder Buffers
  - How Much to Speculate
  - Speculating through Multiple Branches
  - Value Prediction

# **Increasing Instruction Fetch Bandwidth**

- Predicts next instruct address, sends it out *before* decoding instructuction
- PC of branch sent to BTB
- When match is found, Predicted PC is returned
- If branch predicted taken, instruction fetch continues at Predicted PC





2007/4/25



© 2007 Elsevier. Inc. All rights reserved.

# **Example**

- Determine the total branch penalty for a branch target buffer. Make the following assumptions about the prediction accuracy and hit rate:
  - Prediction accuracy is 90% (for instructions in the buffer)
  - Hit rate in the buffer is 90% (for branches predicted taken)

# Example

- We compute the penalty by looking at the probability of two events:
  - The branch is predicted taken but ends up being not taken
  - The branch is taken but is not found in the buffer
- Both carry a penalty of 2 cycles.

Probability(branch in buffer, but actually not taken) = % buffer hit rate x % incorrect predictions =  $90\% \times 10\% = 0.09$ 

Probability(branch not in buffer, but actually taken)= 10%

Branch penalty =  $(0.09 + 0.10) \times 2 = 0.38$ 

# **IF BW: Return Address Predictor**

**Aisprediction frequency** 

- Small buffer of return addresses acts as a stack
- Caches most recent return addresses
- Call ⇒ Push a return address on stack
- Return ⇒ Pop an address off stack & predict as new PC

If the cache is sufficiently large (i.e., as large as the maximum call depth), it will predict the returns perfectly.



2007/4/25
### **Integrated Instruction Fetch Units**

- Integrated branch prediction branch predictor is part of instruction fetch unit and is constantly predicting branches
- Instruction prefetch Instruction fetch units prefetch to deliver multiple instruct. per clock, integrating it with branch prediction
- Instruction memory access and buffering Fetching multiple instructions per cycle:
  - May require accessing multiple cache blocks (prefetch to hide cost of crossing cache blocks)
  - Provides buffering, acting as on-demand unit to provide instructions to issue stage as needed and in quantity needed

# **Speculation: Register Renaming vs. ROB**

- Alternative to ROB is a larger physical set of registers combined with register renaming
  - Extended registers replace function of both ROB and reservation stations
- Instruction issue maps names of architectural registers to physical register numbers in extended register set
  - On issue, allocates a new unused register for the destination (which avoids WAW and WAR hazards)
  - Speculation recovery easy because a physical register holding an instruction destination does not become the architectural register until the instruction commits
- Most Out-of-Order processors today use extended registers with renaming

2007/4/25

# **How Much to Speculate**

- Speculation is not free:
  - It takes time and energy, and the recovery of incorrect speculation further reduces performance
  - The processor must have additional resources, which take silicon area and power
  - If speculation causes an exceptional event to occur, such as a cache or TLB miss, the potential for significant performance loss increase (if that event would not have occurred without speculation)

# How Much to Speculate (cont.)

- To maintain most of the advantage, while minimizing the disadvantages:
  - Most pipelines with speculation will allow only low cost exceptional events (such as a first-level cache miss) to be handled in speculative mode.
  - If an expensive exceptional event occurs, such as a second-level cache miss or a TLB miss, the processor will wait until the instruction causing the event is no longer speculative before handling the event.

#### **Speculating through Multiple Branches**

- Three different situations can benefit from speculating on multiple branches simultaneously:
  - A very high branch frequency
  - Significant clustering of branches
  - Long delays in functional units
- As of 2005, no processor has yet combined full speculation with resolving multiple branches per cycle.

# **Value Prediction**

- Attempts to predict value produced by instruction
  - E.g., Loads a value that changes infrequently
  - an instruction produces a value chosen from asmall set of potential values
- Value prediction is useful if it significantly increases ILP
  - Focus of research has been on loads; so-so results, no processor uses value prediction
    - » The load returns a value that matches the value on the last execution of the load: 5%~80% (SPEC CPU2000)
    - » The load to match any of the most recent 16 values returned: 80%
- Because of the high costs of misprediction and the likely case that misprediction rates will be significant (20% to 50%), researches have focused on accessing which loads are more predictable and only attempting to predict those.
- So-so results, no commercial processor has included value prediction.

# Value Prediction (cont.)

- Related topic is address aliasing prediction
  - RAW for load and store or WAW for 2 stores
- Address alias prediction is both more stable and simpler since need not actually predict the address values, only whether such values conflict
  - Has been used by a few processors

Putting It All Together: The Intel Pentium 4

## **Execution Trace Cache**

- The Pentium 4 uses a novel execution trace cache to generate the uop instruction stream.
- Hold sequences of instructions to be executed including nonadjacent instructions separated by branches (with its own branch target buffer, which predicts the outcome of uop branches).
- Try to exploit the temporal sequencing of instruction execution rather than the spatial locality exploited in a normal cache.

# **Execution Trace Cache**

- By filling the pipeline from the execution trace cache, the Pentium 4 avoids the need to redecode IA-32 instructions whenever the trace cache hits.
- When a trace-cache miss occurred, IA-32 instructions are fetched from the L2 cache and decoded to refill the execution trace cache.
  - Up to 3 IA-32 instructions may be decoded and translated every cycle, generating up to six uops (micro-operations).
  - When a single IA-32 instruction requires more than three uops, the uops sequence is generated from the microcode ROM.

## **Out-of-Order Speculative Pipeline**

- Each clock cycle
  - 3 uops can be renamed and dispatched to the functional unit queues
  - 3 uops can be committed
  - 6 uops can be dispatched to the functional units (4 dispatch ports: load/store units, basic ALU operations, FP and integer operations)











© 2007 Elsevier, Inc. All rights reserved.





#### **Deeper Pipeline**

- The Pentium 4 introduced a much deeper pipeline to achieve a higher clock rate.
- Initial Pentium 4 (introduced in 1990)
  - Minimum # of cycles to transit the pipeline was 21
  - 1.5 GHz clock rate
- Pentium 4 (2004 version)
  - A simple instruction take 31 clock cycles
  - 3.2 GHz clock rate

# **Deeper Pipeline**

- With such deep pipelines and aggressive clock rates, the cost of cache miss and branch mispredictions are both very high.
- A two-level cache is used to minimize the frequency of DRAM accesses.
- Branch prediction is done with a branch-target buffer using a two-level predictor with both local and global histories.
  - The size of the branch-target buffer was increased.
  - The static predictor (used when branch-target buffer misses) was improved.

# **Performance Analysis**

- The processor is a Pentium 4 640 running at 3.2GHz with an 800MHz system bus and 667MHz DDR2 DRAMs for main memory.
- Focus on branch prediction and cache misses
  - Branch-prediction accuracy is crucial in speculative processors, since incorrect speculation requires recovery time and wastes energy pursing the wrong path.
  - The miss penalty for L2 is comparably higher than L1, and the inability of the microarchitecture to hide these very long misses means that L2 misses likely are responsible for an equal of greater performance loss.

#### **Branch Misprediction**



#### **Misspeculation Percentage**



#### L1 and L2 Data Cache Misses



- •The scale of the L1 misses is 10 times that of the L2 misses.
- The miss rate for L1 is about 14 times higher than the miss rate for L2.

#### The CPI for the 10 SPEC CPU Benchmarks



#### The CPI for the 10 SPEC CPU Benchmarks



#### **Fallacies and Pitfalls**

- Fallacy: Processors with lower CPIs will always be faster.
- Fallacy: Processors with faster clock rates will always be faster.
  - Although a lower CPI is certainly better, sophisticated multipleissue pipelines typically have slower clock rates than processors with simple pipelines.
  - In applications with limited ILP or where the parallelism cannot be exploited by the hardware resources, the faster clock rate often wins.
  - When significant ILP exists, a processor that exploits lots of ILP may be better.

#### **Fallacies and Pitfalls**

#### IBM Power5

- Two processor cores each capable of sustaining 4 instructions per clock (2 FP & 2 loadstore instructions)
- The highest clock rate in
  The highest clock rate in 2005 is 1.9 GHz

#### Intel Pentium 4

- A single processor with multithreading. The processor can sustain 3 instructions per clock with a very deep pipeline.
- 2005 is 3.8 GHz

Power5 is faster by 1.5 on SPECfp2000 and the Pentium 4 will be faster by 1.3 on SPECint2000.

#### **Perspective**

- Interest in multiple-issue because wanted to improve performance without affecting uniprocessor programming model.
- Taking advantage of ILP is conceptually simple, but design problems are amazingly complex in practice.
- Conservative in ideas, just faster clock and bigger.

#### **Perspective**

- Processors of last 5 years (Pentium 4, IBM Power5, AMD Opteron) have the same basic structure and similar sustained issue rates (3 to 4 instructions per clock) as the 1st dynamically scheduled, multipleissue processors announced in 1995
  - Clocks 10 to 20X faster, caches 4 to 8X bigger, 2 to 4X as many renaming registers, and 2X as many load-store units

 $\Rightarrow$  performance 8 to 16X

• Peak vs. delivered performance gap increasing

# In Conclusion ...

- Interrupts and Exceptions either interrupt the current instruction or happen between instructions
  - Possibly large quantities of state must be saved before interrupting
- Machines with precise exceptions provide one single point in the program to restart execution
  - All instructions before that point have completed
  - No instructions after or including that point have completed
- Hardware techniques exist for precise exceptions even in the face of out-of-order execution!

Important enabling factor for out-of-order execution