Superscalar Processors

1. Introduction
2. Parallel decoding
3. Superscalar instruction issue
4. Shelving
5. Register renaming
6. Parallel execution
7. Preserving the sequential consistency of instruction execution
8. Preserving the sequential consistency of exception processing
9. Case studies of superscalar processors
Specific tasks of superscalar processing

- Parallel decoding
- Superscalar instruction issue
- Parallel instruction execution
- Preserving sequential consistency of execution
- Preserving sequential consistency of exception processing
Parallel Decoding

Scalar (pipelined)

Superscalar
Specific tasks of superscalar processing

- Parallel decoding
- Superscalar instruction issue
  - Issue policy
  - Issue rate
    - 2
    - 3
    - 4
    - 6
- Parallel instruction execution
- Preserving sequential consistency of execution
- Preserving sequential consistency of exception processing

How and when to send the instruction(s) to EU(s)
Design Space: Issue Policy

- Issue policy
  - False data dependency
    - No extra provision
      - Register renaming
  - Unresolved control dependency
    - Waiting for resolution
  - Use of shelving
    - Blocking issue
    - Shelved issue
  - Handling of issue blockages
    - Preserve issue order
    - Alignment of instruction issue
      - In-order issue
      - Out-of-order issue
      - Aligned issue
      - Unaligned issue
Design Space: Register Renaming

- presumes three-operand format
Scope of Register Renaming

Partial renaming
Renaming is restricted to particular instruction types

*Power1*¹ (RS/6000, 1993)
*Power 2*² (1993)
*PowerPC 601*³ (1993)
*Nx586*⁴ (1994)

Full renaming
Renaming comprises all eligible instruction types

*PowerPC 603* (1993)
*PowerPC 620* (1996)
*R10000* (1996)
and
most recent superscalar processors except the α-line and Sun’s UltraSparc (1995)

The Power1 renames only FP loads
The Power2 extends renaming to all FP instructions
The PowerPC 601 renames only the Link and Count registers
Since the Nx586 is an FX processor, it renames only FX instructions
Type of Rename Buffers

Type of rename buffers
(The basic approach to how rename buffers are implemented)

- Merged architectural and rename register file
  - Power1 (1990)
  - Power2 (1993)
  - ES/9000 (1992p)
  - Nx586 (1994)
  - PM1 (Sparc64, 1995)
  - R10000 (1996)

- Separate rename and architectural register files
  - PowerPC 503 (1993)
  - PowerPC 504 (1995)
  - PowerPC 520 (1996)

- Holding renamed values in the ROB
  - PentiumPro (1995)
  - Am 29000 sup (1995)

- Holding renamed values in the DRIS
  - Lightning (1991p)

DRIS = Deferred scheduling, Register renaming, Instruction Shelf

ROB = Reorder Buffer
## Number of Rename Buffers

<table>
<thead>
<tr>
<th>Processor type</th>
<th>Implementation of renaming</th>
<th>Number of rename buffers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FX</td>
<td>FP</td>
</tr>
<tr>
<td>Merged rename and arch. register file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power1</td>
<td>(1990)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES/9000</td>
<td>(1992p)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(16 arch. + 16 ren.)</td>
</tr>
<tr>
<td>PM1</td>
<td>(1995)</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(78 arch. + 38 ren.)</td>
</tr>
<tr>
<td>R10000</td>
<td>(1996)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(32 arch. + 32 rename)</td>
</tr>
<tr>
<td>Separate rename register file</td>
<td></td>
<td>n.a.</td>
</tr>
<tr>
<td>PowerPC 603</td>
<td>(1993)</td>
<td>12</td>
</tr>
<tr>
<td>PowerPC 604</td>
<td>(1995)</td>
<td>8</td>
</tr>
<tr>
<td>PowerPC 620</td>
<td>(1996)</td>
<td></td>
</tr>
<tr>
<td>Renaming within the ROB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Am29000 sup</td>
<td>(1995)</td>
<td>10</td>
</tr>
<tr>
<td>K5</td>
<td>(1995)</td>
<td>16</td>
</tr>
<tr>
<td>PentiumPro</td>
<td>(1995)</td>
<td>40</td>
</tr>
</tbody>
</table>
Rename Buffers: Access Mechanism
Basic Implementation Alternatives

Renaming implementation alternatives
Design Space: Issue Policy

- False data dependency
  - No extra provision
    - Register renaming
  - Unresolved control dependency
    - Waiting for resolution
    - Speculative branch processing
- Use of shelving
  - Blocking issue
  - Shelved issue
- Handling of issue blockages
  - Preserve issue order
  - In-order issue
    - Out-of-order issue
  - Alignment of instruction issue
    - Aligned issue
    - Unaligned issue
Branch Prediction

Static prediction
- conditional jump would not be taken
- backward branches will be taken, and forward-pointing branches will not be taken

Dynamic prediction
- requires h/w to maintain history
  - 2 bit saturating counter (good for loops)
  - 2-level adaptive predictor (good for recurring patterns)
  - ...

Predication (if structures)
- change the control dependency to a data dependency
Design Space: Shelving

- **Scope of shelving**
  - Partial shelving
  - Full shelving

- **Type**
  - Standalone (Reservation stations)
  - Combined: shelving, renaming, reordering (ROB, DRIS)

- **Layout**
  - Number of shelving buffer entries
  - Number of read and write ports

- **Operand fetch policy**
  - Issue bound fetch
  - Dispatch bound fetch

- **Instruction dispatch scheme**
  - Dispatch policy
  - Dispatch rate
  - Scheme for checking availability of operands
  - Treatment of an empty reservation station
Principle of Shelving

In the absence of hardware constraints, instructions will be issued to shelving buffers despite dependencies.

Instructions wait here until dependencies are resolved.

Instructions are checked for dependencies. An independent instruction is forwarded to the associated EU.
Scope of Shelving

**Partial shelving**

- Shelving is restricted to one or to a few instruction types
  - A few superscalar processors, such as
    - Power1 (1990)
    - Power2 (1993)
      (both processors shelve only FP instructions)
    - MC 88110 (1993)
      (shelves only stores and conditional branches)
    - R8000 (1994)
      (shelves only FP instructions)

**Full shelving**

- Shelving covers all instruction types
  - Most recent superscalar processors, such as
    - PowerPC 603 (1993)
    - PowerPC 604 (1995)
    - PowerPC 620 (1996)
    - Pentium Pro (1995)
    - R10000 (1996)

Issue performance, trend
Type of Shelving Buffer

- **standalone** (reservation station)
  - individual RS
    - RS
    - ... RS
    - EU
    - EU
  - group RS
    - RS
    - EU
    - EU
  - central RS
    - RS
    - EU
    - EU
    - EU

- **combined**: shelving, renaming, reordering
  - DRIS
    - (deferred scheduling, register renaming, instruction shelf)

- **Power 1**: AM2900
- **Power 2**: R1000, Sparc64
- **Pentium Pro**
Number of shelving buffer entries

<table>
<thead>
<tr>
<th>Processor</th>
<th>Total number of shelves</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerPC 603 (1993)</td>
<td>3</td>
</tr>
<tr>
<td>PowerPC 604 (1994)</td>
<td>12</td>
</tr>
<tr>
<td>PowerPC 620 (1995)</td>
<td>15</td>
</tr>
<tr>
<td>Nx586 (1994)</td>
<td>42</td>
</tr>
<tr>
<td>K5 (1995)</td>
<td>14</td>
</tr>
<tr>
<td>PM1 (Sparc64) (1995)</td>
<td>36</td>
</tr>
<tr>
<td>PentiumPro (1995)</td>
<td>20</td>
</tr>
<tr>
<td>R10000 (1996)</td>
<td>48</td>
</tr>
</tbody>
</table>
Operand Fetch Policy

- **Issue bound fetch**
  - Operands are fetched during instruction issue
  - Then shelving buffers hold source operand values
    - IBM 360/91 (1967)
    - PowerPC 603 (1993)
    - PowerPC 604 (1995)
    - PowerPC 620 (1996)
    - PM-1 (Sparc 64) (1995)
    - PentiumPro (1995)

- **Dispatch bound fetch**
  - Operands are fetched during instruction dispatch
  - Then shelving buffers hold source register numbers
    - CDC 6600 (1994)
    - Power1 (1990)
    - Power2 (1993)
    - Lightning (1991p)
    - ES/9000 (1992p)
    - Nx586 (1994)
    - PA 8000 (1996)
    - R10000 (1996)
Operand Fetch during Instruction Issue
Operand Fetch during Instruction Dispatch
Operand Fetch during Instruction Issue
Operand Fetch during Instruction Dispatch
Basic Implementation Alternatives

Renaming implementation alternatives

- Merged architectural and rename register file
  - Access via mapping table
  - Associative access
- Separate rename register file
  - Access via mapping table
  - Associative access
- Renaming within the ROB
  - Access via mapping table
  - Associative access
- Renaming within the DRIS
  - Access via mapping table
  - Associative access

Proposals:
- Keller (1975)
- Smith-Pleszkun (1988)
- Sohi and Vajapeyam (1987)

Processors:
- E/S9000 (1992p)
- PMI (1995)
- Power1 (1990) (Sparc 64)
- Power2 (1993)
- N×586 (1994)
- R10000 (1996)
- Power PC 603 (1993)
- PentiumPro (1995)
- Power PC 604 (1995)
- Power PC 620 (1996)
- Am29000 sup (1995)
- Lightning (1991p)
Most frequently used combinations
Instruction Dispatch

Dispatch order

- In-order dispatch
  - Shelving buffer
  - Check
  - A non-executable instruction blocks the dissemination of subsequent instructions

- Partially out-of-order dispatch
  - Shelving buffer
  - Check
  - Non-executable instructions belonging to particular instruction types block the dissemination of all other subsequent instructions, but others do not

- Out-of-order dispatch
  - Shelving buffer
  - Check
  - A non-executable instruction does not block the dissemination of subsequent executable instructions

Performance, trend

Power1 (1990) \rightarrow Power2 (1993)^1
PowerPC 620 (1996)^3
Nex586 (1994)
Am29000 sup. (1995)
360/91 (1967)
Lightning (1991)
ES/9000 (1992c)
PM1 (1995)
PentiumPro (1995)
R10000 (1996)
PA 6000 (1996)

^1 In the Power2, only a single pending (not executable) FP instruction can be skipped.

^2 Out-of-order dispatch from the three integer reservation stations, but in-order dispatch from the Branch, Load/Store and FP reservation stations.

^3 Out-of-order dispatch from the three integer and Load/Store reservation stations, in-order dispatch from the Branch and FP reservation stations.
Checking availability of operands

Schemes for checking the availability of operands

Direct check of the scoreboard bits

The availability of source operands is not explicitly indicated in the RS. Thus, the scoreboard bits are tested for availability

Usually employed if operands are fetched during instruction dispatch,

Check of the explicit status bits

The availability of source operands is explicitly indicated in the RS. These explicit status bits are tested for availability

Usually employed if operands are fetched during instruction issue,
Checking availability of operands

**Diagram:**
- Decoded instructions:
  - OC, R_{S1}, R_{S2}, R_D
  - Reservation Station
  - Check V-bits of source operands
  - OC, O_{S1}, O_{S2}, R_D
  - EU
  - Result, R_D

**Dispatch:**
- Reset V-bit of R_D
- Update R_D, set V-bit
- R_{S1}, R_{S2}, R_D
- Occ, O_{S1}, O_{S2}, R_D
- EU
- Result, R_D

**Issue:**
- Update R_D, set V-bit
- R_{S1}, R_{S2}, R_D
- Reset V-bit of R_D
- V
- V_{S1}, V_{S2}
- Result, R_D

**Variables:**
- OC: Operation code
- R_{S1}, R_{S2}: Source register numbers
- R_D: Destination register number
- O_{S1}, O_{S2}: Source operand values
- I_{S1}, I_{S2}: Source operand identifiers (tags)
- V_{S1}, V_{S2}: Source operand valid bits
- RS: Reservation station

**Legend:**
- Reg. file
- Scoreboard bits
- Reservation Station
Design Space: Issue Policy

- False data dependency
  - No extra provision
    - Register renaming
- Unresolved control dependency
  - Waiting for resolution
- Use of shelving
  - Blocking issue
    - Shelved issue
- Handling of issue blockages
  - Preserve issue order
  - In-order issue
  - Out-of-order issue
  - Alignment of instruction issue
    - Aligned issue
    - Unaligned issue
Issue order of instructions

Preserving issue order

In-order issue

Out-of-order issue

Instructions to be issued

Issue window

e d c b a

Instructions issued

Instructions are issued strictly in program order

Instructions may be issued out of order

Designates an independent instruction

Designates a dependent instruction

Designates an issued instruction
Alignment of Instruction Issue

* example assumes “in order” issue
Alignment of Instruction Issue

Instructions are issued from a fixed window
- Typical for first-generation superscalar processors and for recent processors using shelving, e.g.,
  - i960CA (1989)
  - Power1 (1990)
  - PA 7100 (1992)
  - SuperSparc (1992)
  - PowerPC 603 (1993)
  - PowerPC 604 (1995)
  - PowerPC 620 (1996)
  - PA 8000 (1996)
  - R10000 (1996)
  - PM1 (Sparc64) (1995)
  - but also in the
    - a 21064 (1992)
    - a 21064A (1994)
    - a 21164 (1995)

Instructions are issued from a gliding window
- Typical for follow-on superscalar processors which do not use shelving, e.g.,
  - MC 88110 (1993)
  - MC 68060 (1993)
  - PA 7100LC (1993)
  - R8000 (1994)
  - PA 7200 (1995)
  - UltraSparc (1995)

- Designates an independent instruction
- Designates a dependent instruction. Here, for simplicity, we assume that a dependent instruction becomes dependency free in the next cycle after the preceding instruction has been issued
- Designates an issued instruction
Specific tasks of superscalar processing

- Parallel decoding
- Superscalar instruction issue
- Parallel instruction execution
- Preserving sequential consistency of execution
- Preserving sequential consistency of exception processing
Parallel Execution

- Executing several instructions in parallel
  - instructions will generally finish in out-of-program order
- to finish
  - operation of the instruction is accomplished, except for writing back the result into
    - the architectural register or
    - memory location specified, and/or
    - updating the status bits
- to complete
  - write back the results
- to retire (ROB)
  - write back the results, and
  - delete the completed instruction from the last ROB entry
Preserving Sequential Consistency of Instruction Execution

For multiple EUs operating in parallel, the overall instruction execution should mimic sequential execution

- the order in which instructions are completed
- the order in which memory is accessed
Sequential consistency of instruction execution

Processor consistency

Consistency of the sequence of instruction completions

Weak processor consistency
- Instructions may complete out-of-order, provided that no dependencies are adversely affected
- Instruction reordering is allowed
- Detection and resolution of dependencies ensures weak processor consistency
  - Power1 (1990)
  - Power2 (1993)
  - MC88110 (1993)
  - PowerPC 601 (1993)
  - UltraSPARC (1995)
  - PM1 (1995)
  - Am29000 sup (1995)
  - PA 8000 (1996)
  - R10000 (1996)

Strong processor consistency
- Instructions complete strictly in program order
- No instruction reordering is allowed
- ROB ensures strong processor consistency

Memory consistency

Consistency of the sequence of memory accesses

Weak memory consistency
- Memory accesses due to load and store instructions may be out-of-order, provided that no dependencies are adversely affected
- Load/store reordering is allowed
- Detection and resolution of memory data dependencies ensures weak memory consistency
  - MC88110 (1993)
  - PowerPC 602–620
  - UltraSPARC (1995)
  - PM1 (1995)
  - PA 8000 (1996)
  - R10000 (1996)

Strong memory consistency
- Memory is accessed due to load and store instructions strictly in program order
- No load/store reordering is allowed
- The ROB may be used to ensure strong memory consistency

Trend

Trend, performance
Preserving the Sequential Consistency of Exception Processing

- When instructions are executed in parallel,
  - interrupt requests, which are caused by exceptions arising from instruction execution, are also generated out of order.

- If the requests are acted upon immediately,
  - the requests are handled in different order than in a sequential operation processor and you have imprecise interrupts

- Precise interrupts: handling the interrupts is consistent with the state of a sequential processor
Sequential Consistency of Exception Processing

Sequential consistency of exception processing

Weak consistency

Imprecise interrupts

*Power1 (1990)*
*Power2 (1993)*
*
aprocessors

Strong consistency

Precise interrupts

*MC88110 (1993)*
*Pentium (1993) and usually processors making use of an ROB, such as:*
*ES/9000 (1992p)*
*PowerPC line*
*PA 8000 (1996)*
*R10000 (1996)*
Instruction issue policies of superscalar processors

- **Straightforward superscalar issue**
  - A: No renaming
  - B: Speculative execution
  - C: Blocking issue

- **Straightforward superscalar issue with shelving**
  - A: No renaming
  - B: Speculative execution
  - C: Shelved issue

- **Straightforward superscalar issue with renaming**
  - A: Renaming
  - B: Speculative execution
  - C: Blocking issue

- **Advanced superscalar issue**
  - A: Renaming
  - B: Speculative execution
  - C: Shelved issue

**Aligned issue**
Typical in early first-generation superscalar proc.

- **Alignment-free issue**
  Typical in follow-on first-generation superscalar proc.
  - MC 68060 (1993)
  - MC 88110 (1993)

- R8000 (1994)

**Issue performance, trend**

Instruction issue policies

- Traditional scalar issue
  - No renaming
  - No speculative execution
  - Blocking issue
  - Scalar issue

- Traditional scalar issue with speculative execution
  - No renaming
  - Speculative execution
  - Blocking issue
  - Scalar issue

- Straightforward superscalar issue
  - No renaming
  - Speculative execution
  - Blocking issue
  - Superscalar issue

- Advanced superscalar issue
  - Renaming
  - Speculative execution
  - Shelved issue
  - Superscalar issue

Typical in traditional processors and in early pipelined microproc.

- i86 (1978)
- i286 (1982)
- i386 (1985)

- MC 68000 (1979)
- MC 68020 (1982)
- MC 68030 (1987)

- R2000 (1987)
- R3000 (1989)

- CY7C601 Sparc (1991)

Typical in follow-on pipelined microproc.

- i486 (1988)
- Pentium (1993)
- PS (1995)

- MC 68040 (1990)
- MC 68060 (1993)

- R4000 (1992p)
- R8000 (1994)
- R10000 (1996)

- PA 7100 (1992)
- PA 7200 (1995)
- PA 6000 (1996)

Alignment-free issue

- Typical in early first-generation superscalar proc.
- Typical in follow-on first-generation superscalar proc.

- PowerPC 601 (1993)
- PowerPC 603 (1993)
- PowerPC 604 (1994)
- PowerPC 620 (1995)

- UltraSparc (1995)
- PM1 (Sparc64) (1995)

- SuperSparc (1992)
- PA 6000 (1996)

- PowerPC 604 (1994)
- PowerPC 620 (1995)

Performance, trend

- a21064 (1993)
- a21064A (1994)
- a21164 (1995)