Optimal pipeline depth

Anshuman Gupta
### Timing Path

**Startpoint:** router/need_out_1_3_reg

**Endpoint:** router/fifo_3_8/data/0/ei_out_r2

**Path Group:** my_clock

**Path Type:** net

<table>
<thead>
<tr>
<th>Point</th>
<th>Incr</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock my_clock (rise edge)</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>clock network delay (ideal)</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>router/need_out_1_3_reg/CK (OFFH000TH)</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>router/need_out_1_3_reg/Q (OFFH000TH)</td>
<td>0.0086</td>
<td>0.0086</td>
</tr>
<tr>
<td>router/25/Y (CLKIN32768TH)</td>
<td>0.0086</td>
<td>0.0092</td>
</tr>
<tr>
<td>router/27/Y (CLKIN32768TH)</td>
<td>0.0082</td>
<td>0.1393</td>
</tr>
<tr>
<td>router/13/Y (NONCLK6TH)</td>
<td>0.0062</td>
<td>0.1989</td>
</tr>
<tr>
<td>router/16/Y (NONCLK6TH)</td>
<td>0.0062</td>
<td>0.2256</td>
</tr>
<tr>
<td>router/24/Y (CLKIN32768TH)</td>
<td>0.0028</td>
<td>0.3533</td>
</tr>
<tr>
<td>router/12/Y (NONCLK6TH)</td>
<td>0.0024</td>
<td>0.3384</td>
</tr>
<tr>
<td>router/12/Y (NONCLK6TH)</td>
<td>0.0016</td>
<td>0.4952</td>
</tr>
<tr>
<td>router/11/Y (AND000TH)</td>
<td>0.0016</td>
<td>0.4572</td>
</tr>
<tr>
<td>router/fifo_3_8/data/seqc (/fifo_data_6)</td>
<td>0.0000</td>
<td>0.5752</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0000</td>
<td>0.5752</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0000</td>
<td>0.5752</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0000</td>
<td>0.5752</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0033</td>
<td>0.6146</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0012</td>
<td>0.6050</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0032</td>
<td>0.7436</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0032</td>
<td>0.7729</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/datac (/fifo_data_6)</td>
<td>0.0025</td>
<td>0.8014</td>
</tr>
<tr>
<td>router/fifo_3_8/data/0/ei_out_r2/CK (OFFH000TH)</td>
<td>0.0000</td>
<td>0.8014</td>
</tr>
</tbody>
</table>

**Data Arrival Time:** 0.0014

**Clock (NET):** 0.0029
Experiment Setup

Calculate the useful work in each stage (FO4)

Assume naive pipelining for each stage

Calculate cycle time using -

\[ \phi = \phi_{\text{logic}} + \phi_{\text{latch}} + \phi_{\text{skew}} + \phi_{\text{jitter}} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_{\text{latch}} )</td>
<td>Latch Overhead</td>
<td>1.0 FO4</td>
</tr>
<tr>
<td>( \phi_{\text{skew}} )</td>
<td>Skew Overhead</td>
<td>0.3 FO4</td>
</tr>
<tr>
<td>( \phi_{\text{jitter}} )</td>
<td>Jitter Overhead</td>
<td>0.5 FO4</td>
</tr>
<tr>
<td>( \phi_{\text{overhead}} )</td>
<td>Total</td>
<td>1.8 FO4</td>
</tr>
</tbody>
</table>

Assume \( \phi_{\text{overhead}} \) remains the same over generations
Figure 5: The harmonic mean of the performance of integer and floating point benchmarks, executing on an out-of-order pipeline, accounting for latch overhead, clock skew and jitter. For integer benchmarks best performance is obtained with 6 FO4 of useful logic per stage ($\phi_{\text{logic}}$). For vector and non-vector floating-point benchmarks the optimal $\phi_{\text{logic}}$ is 4 FO4 and 5 FO4 respectively.
Sensitivity to $\Phi_{overload}$

![Graph showing sensitivity to $\Phi_{overload}$]
Sensitivity to micro-architectural loops

The graph shows the relative Instruction Per Cycle (IPC) over the number of cycles for different conditions over the Alpha 21264 loop. The conditions include:

- load-use
- branch mis-pred
- issue-wakeup

The graph indicates that the relative IPC decreases as the number of cycles increases, with the issue-wakeup condition showing the most significant decrease, followed by branch mis-pred, and then load-use.
Reclaiming lost performance

- Architectural loops cause heavy IPC hits with increase in pipeline stages
- Reduce the impact of increased latency to reduce IPC losses
- Use locality, temporality and criticality
Segmented Instruction Window

[Diagram description]

New Instructions → Stage 4 (Tag Latch, Tag Latch) → S4 → L7 (1 Instruction) → S3 → L5-16 → S2 → L0-L4 → IS (4 Instructions) → Selected Instructions

- Stage 1 (Tag Latch, Tag Latch) → 8 Instructions

Destination Tags

[Graph description]

Relative IPC vs. Instruction window pipeline depth:
- vector FP
- Integer
- Non-vector FP
Power kicks in
Optimal Power/Performance Pipeline Depth

Performance Equation

\[
\frac{T}{N_I} = \left( \frac{t_o}{\alpha} + \frac{\gamma N_H}{N_I} t_p \right) + \frac{t_p}{\alpha p} + \frac{\gamma N_H t_o}{N_I} p
\]

- Time taken is proportional to -
  - \( N_H \) - number of hazards
  - \( \gamma \) - average stalling ratio due to hazards
  - \( t_o \) - overhead delay
  - \( t_p \) - total useful pipeline time

- Time taken goes down with -
  - \( \alpha \) - degree of superscalar execution

- The relation to number of pipeline stages is not linear
Optimal Power/Performance Pipeline Depth

Power Equation

\[ P_T = (f_{cg} f_s P_d + P_l) N_L p^\eta \]

- \( P_d \) - dynamic power, based on
- \( f_{cg} \) - clock gating degree
- \( f_s \) - frequency
- \( P_l \) - leakage power
- \( N_L \) - number of latches
- \( p \) - number of pipeline stages

The growth with number of pipeline stages is super-linear due to growth in number of latches by factor \( \eta \).
Theoretical results

- $p_{\text{opt}}$ - optimal number of pipeline stages

- $p_{\text{opt}} \propto N_H$

- $p_{\text{opt}} \propto \gamma$

- $p_{\text{opt}} \propto \alpha$

- $p_{\text{opt}} \propto t_p/t_0$
Simulation results

![Graph showing simulation results for different values of m (1, 2, 3) with BIPS]
Simulation results
Conclusions

- Power matters!!!
- These results hold true in future only if no major technological breakthrough achieved for changing the equations
- Hazards ($N_H$) can’t be reduced but their impact ($\gamma$) can be reduced, traditionally a major research area
- Superscalar architectures prefer shorter pipelines
- Like NUCA we are soon going to see varying access times to L1, register files etc.
APPENDIX slides
A flip flop is created by using 2 latches (master-slave pair)