Early Vector Computing

John G. Zabolitzky

www.cray-cyber.org

www.gfhr.de
High Performance Scalar

• CDC 7600 (1968) marks the limit of pipelined scalar execution
• RISC invention by Seymour Cray
• fetch, decode, execute overlapped between instructions, optimally one cycle each
• maximum performance: 1 instruction/cycle
• 36 MHz => 36 MIPS => 5 MFLOPS
# Pipelined Scalar Execution

<table>
<thead>
<tr>
<th>instruction</th>
<th>time</th>
<th>fetch</th>
<th>decode</th>
<th>execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>fetch</td>
<td>decode</td>
<td>execute</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>fetch</td>
<td>decode</td>
<td>execute</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>fetch</td>
<td>decode</td>
<td>execute</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>fetch</td>
<td>decode</td>
<td>execute</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>fetch</td>
<td>decode</td>
<td>execute</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>fetch</td>
<td>decode</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>fetch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pipelined execution on parallel functional units
Scalar Code Example

• \textbf{DO } i=1,100 \quad a(i)=b(i)\times c(i)
  
  \begin{itemize}
  \item load b, inc address
  \item load c, inc address
  \item multiply
  \item store a, inc address
  \item decrement count, loop?
  \end{itemize}

• 5 instructions = cycles (optimum) for one multiply

• pipelined multiply: could start one multiply each and every cycle => only 20\% efficient use
Computer Hardware Real Estate

• Memory 50%
• CPU 50%
  – Multiplier 50% (64 bit ~ 4096 elements)
  – all other 50%
• Goal: keep multiplier busy all the time
  – most cost-effective computer
  – most reliable (least components) at same performance
Architectural Alternatives

• Pipelined Scalar (RISC) as outlined
• Pipelined Vector (this presentation)
• SIMD (Single Instruction Multiple Data) parallel arithmetic (e.g., ILLIAC IV)
• Superscalar = multiple issue in one cycle
  – all modern single-chip CPUs (Intel to TI)
• VLIW (Very Long Instruction Word) variant of superscalar
• MIMD true parallel streams, e.g. Cray T3E
Alternatives Evaluation

- SIMD: too many parts => unreliable, too expensive, low usage percentage
- Superscalar: possible, but difficult c/o conflict resolution, more general than vector
- VLIW: essentially unprogrammable, but lowest hardware cost/performance
- MIMD: does not address this problem, may be superimposed upon ANY CPU structure
Vector Computation

• Scientific codes have high percentage in looping over simple data structures
• DO  i=1,100  a(i) = b*c(i) + d(i)
• simple logical structure ==> 
• set up such that one multiply/cycle
• one instruction for entire loop
• MFLOP rate = cycle rate or multiple thereof
• specialized for scientific/engineering tasks
Vector Pipeline  \[ c(i) = a(i) \times b(i) \]

<table>
<thead>
<tr>
<th>fetch a(i++)</th>
<th>multip. 1</th>
<th>multip. 2</th>
<th>multip. 3</th>
<th>multip. 4</th>
<th>store c(i++)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fetch b(i++)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>V</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
First Vector Computers

- Control Data Corporation (CDC) STAR-100
  - [STring ARray 100 MFLOPS]
  - memory-to-memory architecture
  - therefore long startup times (≈n00 cycles)
  - very slow scalar unit (≈2 MFLOPS)
  - overall disappointing performance
- contracted 1967, announced 1972, delivered 1974
- total of 4 machines, 2 Lawrence Livermore Lab
- Thornton (CDC) and Fernbach (LLL) loose their jobs
CDC STAR-100

Photograph courtesy of Charles Babbage Institute, University of Minnesota, Minneapolis
Texas Instruments ASC

- Advanced Scientific Computer
- architecturally similar to CDC STAR-100
- 7 units sold
- TI dropped out of mainframe computer manufacturing after this machine
Vector Performance I

- MFLOP rate as function of vector length
- scalar: \(~\text{constant (loop overhead)}\)
- vector:
  - \# cycles = startup + length / nflop\_per\_cycle
  - rate/clock = \#ops / \#cycles \sim n / (startup + n)
  - half rate at vector length \( n \sim \text{startup} \)
  - full rate needs \( n \gg \text{startup} \Rightarrow \text{“Long Vector Machine”} \)
Performance vs. Startup, Length

![Performance vs. Startup, Length Graph](image-url)
Vector Performance II

• Vector/Scalar Subsections
  
  – ALL codes have some scalar (non-vectorizable) sections
  
  – total time = (scalar fraction)/(scalar rate) + (vector fraction)/(vector rate)
  
  – example: 10% / 1 MFLOPS + 90% / 100 MFLOPS = 100 / (0.1 * 100 + 0.9 * 1) = 9.2 MFLOPS !!!
Vector Version of Amdahl’s Law

![Graph showing the vector version of Amdahl’s Law with performance on the y-axis and scalar fraction on the x-axis. The graph includes curves for different scalar fractions: r5, r10, r20, r50, and r100.](image-url)
Vector Computer Design Guide

• Must have SHORT vector startup => can work with short vectors
• Must have FASTEST POSSIBLE scalar unit => can afford scalar sections
• irregular data structures ==> need gather, scatter, merge operations (and a few more)
  – \( x(i) = a(\text{index}(i)) \times b(i) \)
  – \( y(\text{index}(i)) = c(i) + d(i) \)
  – where \( (a(i) > b(i)) \) \( c(i) = d(i) \)
Cray Research, Inc.

- Founded by Seymour Cray (father of CDC 6600/7600) in 1972 (STAR-100 known)
- first Cray-1 delivered in 1976 to Los Alamos Scientific Laboratory (LASL)
- 8 vector registers of 64 elements each
- Vector load/store instructions
- fastest scalar computer of its time
- 160 MFLOPS peak rate (2 ops/cycle @ 80 MHz), few cycles startup
Block Diagram Cray YMP-EL, only one of four identical CPUs shown, simplified

Shared

Main

Memory

128 MW

64 bit

1 Gbyte

4 ports/proc

4x 4 x 33 MHz = 4.2 Gby/sec

48 shared registers

8 vector registers, 64 elements, 64 bit

4 vector execution units, 33 MHz

8 scalar registers, 64 bit

Scalar functional units

64 word Tjk
Buffer memory

8 address registers, 32 bit

Address functional units

64 word Bjk
buffer memory

8 instruction buffers, 32 words each

40 Mbyte/sec

Y1 channel

8 instruction issue

IOS
Cray Research, Inc. cnt’d

- 1982 Cray-XMP (Chen improvements)
- 1985 Cray-2, 256 Mword memory, immersion cooled
- 1988 Cray-YMP (last Chen machine)
- 1991 Cray C90
- 1993 Cray T3D (massively parallel Alpha)
  one and only Cray-3 delivered to NCAR (Cray Comp Corp)
- 1994 Cray J90, air cooled
- 1995 Cray T3E, Cray T90 (immersion cooled)
  Cray-4 abandoned (Cray Computer Corporation ch. 11)
- 1996 acquired by Silicon Graphics
- 1998 Cray SV1 air cooled
- 1999 acquired by Teradata => Cray, Inc.
- 2002 Cray X-1, immersion spray cooled
CDC Cyber 200 Family

• 1980, enhanced version of STAR-100
• reduced startup time, ~ 50 cycles
• fast scalar unit
• rich instruction repertoire
• still memory-to-memory, 400 MFLOPS peak
• Cyber 203, Cyber 205, ETA-10 [10 GFLOPS]
• terminated in 1989 since unprofitable
• around 40 Cyber 200, 34 ETA-10 sold
Scientific Computing To-Day

• 1964-1990: all successful scientific computers have been Cray architecture machines (6600/7600/Cray-1/Cray-xyz/Japanese)

• After excursion into massively parallel slow CPUs (e.g., Cray T3E, IBM SPn) we see today the return to enhanced Cray architecture (NEC SX-6 Earth Simulator = 5104 Vector CPUs @ 8 GFLOPs air [100.000 m³/sec] cooled; Cray-X1)

• more cost-conscious detail design