Dataflow
Some Dataflow History

Toffoli=>Fredkin Gate

CAM architecture

PhD Thesis of Norman Margolus, MIT 1987

Today:

CISCO Internet Switch

Data reduction in Physics

Maxeler Dataflow Engines
“Is the LHC throwing away too much data?” New Scientist 2012.
Future of Medicine...

Sequencing

- Cancer: multiple genomes per tissue
- Spain: 28PB for 1 genome per cancer

Modelling

- Max Planck Institute: $10^{18}$ core-hours for Europe
- Cancer, Diabetes, etc...

Big Data

- Healthcare requires complete medical records
- Images, genomes, transcriptome, proteom, ...
Explaining Control Flow versus Data Flow

The Ford Production Line

- Experts are expensive and slow (control flow)
- Many specialized workers are more efficient (data flow)
Technology
resulting from our research at Stanford, Bell Labs, and Imperial College

MAXELER DATAFLOW COMPUTING

One result per clock cycle

Dynamic (switching) Power Consumption:

\[ P_{avg} = C_{load} \cdot V_{DD}^2 \cdot f \]

Minimal frequency \( f \) achieves maximal performance, thus for a given power budget, we get Maximum Performance Computing (MPC)!
Maxeler Architecture Series 2012

Figure 3. MPC-C Series Architecture

Figure 4. MPC-X Series Architecture
Cox-Ingersol (CIR) Interest Rate Model

\[ dr(t) = \alpha(b - r(t))dt + \sigma \sqrt{r(t)}dW(t) \]

\[
r(t_{i+1}) = r(t_i) + \alpha(b(t_i) - r(t_i))[t_{i+1} - t_i] + \sigma \sqrt{r(t_i) + \sqrt{t_{i+1} - t_i}}Z_{i+1}
\]

\[
r(t_{i+1}) = \frac{\sigma^2(1-e^{-\alpha(t_{i+1}-t_i)})}{4\alpha} \chi^2_2 \left( \frac{4\alpha e^{-\alpha(t_{i+1}-t_i)}}{\sigma^2(1-e^{-\alpha(t_{i+1}-t_i)})} r(t_i) \right), \quad d = \frac{4\bar{b}\alpha}{\sigma^2}
\]

\[
B(t, T) = e^{-A(t,T)r(t)+C(t,T)}
\]

\[
A(t, T) = \frac{2(e^{\gamma(T-t)}-1)}{(\gamma + \alpha)(e^{\gamma(T-t)} - 1) + 2\gamma}
\]

\[
C(t, T) = \frac{2\alpha b}{\sigma^2} \log \left( \frac{2\gamma e^{(\alpha+\gamma)(T-t)/2}}{(\gamma + \alpha)(e^{\gamma(T-t)} - 1) + 2\gamma} \right)
\]
• Compute value of complex financial derivatives (CDOs)
• Typically run overnight, but beneficial to compute in real-time
• Many independent jobs
• Speedup: 220-270x
• Power consumption per node drops from 250W to 235W/node
Running with SEG Salt Model

- Running on MaxNode servers
  - 8 parallel compute pipelines per chip
  - 150MHz => low power consumption!
  - 30x faster than microprocessors

Compared to 32 3GHz x86 cores parallelized using MPI

- 15Hz peak frequency
- 30Hz peak frequency
- 45Hz peak frequency
- 70Hz peak frequency

*presented at SEG 2010.

8 Full Intel Racks ~100kWatts => 2 MaxNodes (2U) Maxeler System <1kWatt
Typical Scalability of Sparse Matrix

**Eclipse Benchmark**
(2 node Westmere 3.06 GHz)

**E300 2 Mcell Benchmark**

**Visage – Geomechanics**
(2 node Nehalem 2.93 GHz)

**FEM Benchmark**
Sparse Matrix Solving

O. Lindtjorn et al, 2010

• Given matrix $A$, vector $b$, find vector $x$ in:
  $$Ax = b$$

• Typically memory bound, not parallelisable.

• 1 MaxNode achieved 20-40x the performance of an Intel node.
Economics of Computation (TCO)

**Example Solution 1**
- $50x$ Speed-up per 1U server node
- $32$ Maxeler Node Solution
- Equivalent to $1600$ CPU-only Nodes
- $3.2m$ Operational cost savings over 3 years

**Example Solution 2**
- $30x$ Speed-up per 1U server node
- $40$ Maxeler Node Solution
- Equivalent to $1200$ CPU-only Nodes
- $1.8m$ Operational cost savings over 3 years

**Example Solution 3**
- $20x$ Speed-up per 1U server node
- $50$ Maxeler Node Solution
- Equivalent to $1000$ CPU-only Nodes
- $1.7m$ Operational cost savings over 3 years

**Example Solution 4**
- $40x$ Speed-up per 1U server node
- $32$ Maxeler Node Solution
- Equivalent to $1280$ CPU-only Nodes
- $2.6m$ Operational cost savings over 3 years
Trading: Direct Market Access and HFT
Maxeler Hardware Solutions 2012

CPU plus DFEs
Intel Xeon CPU cores and up to 6 DFEs with 288GB of RAM

DFEs shared over Infiniband
Up to 8 DFEs with 384GB of RAM and dynamic allocation of DFEs to CPU servers

Low latency connectivity
Intel Xeon CPUs and 1-2 DFEs with up to six 10Gbit Ethernet connections

MaxWorkstation
Desktop development system

MaxCloud
On-demand scalable accelerated compute resource, hosted in London
MPC-X Series

1U dataflow cloud providing dynamically scalable compute capability over Infiniband

**MPC-X1000**

- 8 *vectis* dataflow engines (DFEs)
- 192GB of DFE RAM
- Dynamic allocation of DFEs to conventional CPU servers
  - Zero-copy RDMA between CPUs and DFEs over Infiniband
- Equivalent performance to 40-60 x86 servers
MaxCompiler Development Process

CPU
- Host Code
- SLiC
- MaxelerOS

Main Memory

Manager (.java)
Manager m = new Manager();
Kernel k =
    new MyKernel();
m.setKernel(k);
m.setIO(
    link("x", PCIE),
    link("y", DRAM_LINEAR1D));
m.addMode(modeDefault());
m.build();

MyKernel (.java)
HWVar x = io.input("x", hwInt(32));
HWVar result = x * x + 30;
io.output("y", result, hwInt(32));

CPUCode (.c)
#include “MaxSLiCInterface.h”
#include “Calc.max”
int *x, *y;
Calc(x, DATA_SIZE)

Device
- max_open_device(maxfile, "/dev/maxeler0")

Calc(x, DATA_SIZE)
Main
- Memory
- CPU
- Host Code
- SLiC
- MaxelerOS

Manager

Chip

PCI
Express

x
x
+ 30
x
x
+ 30
y

30
Data flow graph as generated by MaxCompiler
4866 nodes; about 250x100
The goal is to maximize utilization of resources on the chip, and bandwidth on the memory bus.

Chip Resource Usage

<table>
<thead>
<tr>
<th>LUTs</th>
<th>FFs</th>
<th>BRAMs</th>
<th>DSPs</th>
<th>: MyKernel.java</th>
</tr>
</thead>
<tbody>
<tr>
<td>727</td>
<td>871</td>
<td>1.0</td>
<td>2</td>
<td>resources used by this file</td>
</tr>
<tr>
<td>0.24%</td>
<td>0.15%</td>
<td>0.09%</td>
<td>0.10%</td>
<td>% of available</td>
</tr>
<tr>
<td>71.41%</td>
<td>61.82%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>% of total used</td>
</tr>
<tr>
<td>94.29%</td>
<td>97.21%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>% of user resources</td>
</tr>
</tbody>
</table>

```java
public class MyKernel extends Kernel {
    public MyKernel (KernelParameters parameters) {
        super(parameters);
        HWVar p = io.input("p", hwFloat(8,24));
        HWVar q = io.input("q", hwUInt(8));
        HWVar offset = io.scalarInput("offset", hwUInt(8));
        HWVar addr = offset + q;
        HWVar v = mem.romMapped("table", addr,
                                hwFloat(8,24), 256);
        p = p * p;
        p = p + v;
        io.output("r", p, hwFloat(8,24));
    }
}
```
Measuring Utilization

- *Top* measures % of time CPU is running
- *Maxtop* monitors % of time the DFE is running

MaxTop Tool 2011.2
Found 2 Maxeler card(s) running MaxelerOS 2011.2
Card 0: MAX3A (P/N: 13424) S/N: 219270088 Mem: 24GB DFE(s): 1 /dev/maxeler0
Card 1: MAX3A (P/N: 13424) S/N: 000025559 Mem: 24GB DFE(s): 1 /dev/maxeler1

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>%DFE</th>
<th>TEMP</th>
<th>BITSTREAM</th>
<th>PID</th>
<th>USER</th>
<th>TIME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxeler0</td>
<td>66.6%</td>
<td>57.1C</td>
<td>9d9de1...</td>
<td>12333</td>
<td>jspooner</td>
<td>00:00:39</td>
<td>model</td>
</tr>
<tr>
<td>maxeler1</td>
<td>0.0%</td>
<td>54.6C</td>
<td>9d9de1...</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
The Exascale Supercomputer (2018)

• 1 exaflop = \(10^{18}\) FLOPS
• Using processor cores with 8FLOPS/clock at 2.5GHz
• 50M CPU cores
• What about power?
  – Assume power envelope of 100W per chip
  – Moore’s Law scaling: 6 cores today \(\rightarrow\) \(\sim100\) cores/chip
  – 500k CPU chips
• 50MW \((\text{just for CPUs!})\) \(\rightarrow\) 100MW likely
• ‘Jaguar’ power consumption: 6MW for 2 Petaflops
Exascale Dataflow Computing

- Exascale application performance by 2018
  - 100 racks
  - <3MW power
- A custom silicon solution could improve this by 5x
  - 20 racks
  - <1MW power
Recent Scientific Publications with Clients

- **Rapid Computation of Value and Risk for Derivatives Portfolios**
  S. Weston†, J. Spooner†, S. Racanière† and O. Mencer‡§.
  †JPMorgan, ‡Maxeler Technologies, §Imperial College London

- **Beyond Traditional Microprocessors for Geoscience High-Performance Computing Applications**
  O. Lindtjorn†, R. G. Clapp†, O. Pell‡, O. Mencer†, M. J. Flynn† and H. Fu§.
  †Stanford University and Schlumberger, ‡Maxeler Technologies, §Tsinghua University

- **Accelerating the Computation of Portfolios of Tranch Credit Derivatives**
  S. Weston†, J-T. Marin†, J. Spooner†, O. Pell† and O. Mencer†.
  †JPMorgan, ‡Maxeler Technologies
  IEEE Workshop on High Performance Computational Finance, New Orleans, USA, November 2010.

- **FD modeling beyond 70Hz with FPGA acceleration**
  D. Oriato†, O. Pell†, C. Andreoletti† and N. Bienati†.
  †Maxeler Technologies, ‡Eni E&P Division
  SEG 2010 HPC Workshop, Denver, USA, October 2010.

- **Surviving the End of Scaling of Traditional Microprocessors in HPC**
  O. Lindtjorn†‡, R. G. Clapp‡, O. Pell§, O. Mencer§ and M. J. Flynn§.
  †Schlumberger, ‡Stanford University, §Maxeler Technologies
  IEEE HOT CHIPS 22, Stanford, USA, August 2010.

- **Fast 3D ZO CRS Stack - An FPGA Implementation of an Optimization Based on the Simultaneous Estimate of Eight Parameters**
  P. Marchetti†, D. Oriato‡, O. Pell‡, A.M. Cristini§ and D. Theis§.
  †Eni E&P Division, ‡Maxeler Technologies, §CRS4
  72nd European Association of Geoscientists and Engineers (EAGE) Conference, Barcelona, June 2010.

- **Computational acceleration of credit and interest rate derivatives**
  O. Mencer† and S. Weston‡.
  †Maxeler Technologies, ‡JPMorgan
MAX-UP: Maxeler University Program

- Founding Members: Imperial, Tsinghua, U Tokyo and Stanford
- MaxAcademy course material for teaching
- MaxTraining at Maxeler: learn dataflow programming
- Internships and collaboration
- Joint research grant applications
- Demos

MaxWorkstation

Brain Network

Earthquake Prediction