Reconfigurable Computing
— Evolution of Von Neumann Architecture

Dr. ShaoJun WEI, Professor
Director, Center for Mobile Computing
Tsinghua University, PRC
December 8, 2010
Outlines

- Introduction
- Von Neumann Architecture
- Reconfigurable Computing
- Major Challenges
- Conclusions
Moore & Von Neumann

Golden Moore

Von Neumann

Scaling-down

Semiconductor

Computer

Von Neumann Architecture

P4004: 1971年世界上第一个微处理器在英特尔公司诞生

Introduction
The experts look ahead

**Cramming more components onto integrated circuits**

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip.

By Gordon E. Moore
Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

"The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the equipment to be used in environments that would otherwise be too hazardous or not practical for use by humans."

With unit cost falling as the number of components per circuit rise, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip.
Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions

ROBERT H. DENNARD, MEMBER, IEEE, FRITZ H. GAENSSEN, HUA-NIEN YU, MEMBER, IEEE, V. LEO RIDEOUT, MEMBER, IEEE, ERNEST BASSOUS, AND ANDRE R. LEBLANC, MEMBER, IEEE

Classic Paper

The successful scaling of integrated circuits depends on the development of new device fabrication processes and materials that allow the continued miniaturization of device features. As devices are scaled down, their gate lengths, gate oxide thicknesses, and channel lengths are reduced to the point where the channel conductance is no longer adequately modeled using conventional semiconductor physics. The successful scaling of integrated circuits also depends on the development of new device fabrication processes and materials that allow the continued miniaturization of device features. As devices are scaled down, their gate lengths, gate oxide thicknesses, and channel lengths are reduced to the point where the channel conductance is no longer adequately modeled using conventional semiconductor physics. The successful scaling of integrated circuits also depends on the development of new device fabrication processes and materials that allow the continued miniaturization of device features. As devices are scaled down, their gate lengths, gate oxide thicknesses, and channel lengths are reduced to the point where the channel conductance is no longer adequately modeled using conventional semiconductor physics. The successful scaling of integrated circuits also depends on the development of new device fabrication processes and materials that allow the continued miniaturization of device features. As devices are scaled down, their gate lengths, gate oxide thicknesses, and channel lengths are reduced to the point where the channel conductance is no longer adequately modeled using conventional semiconductor physics. The successful scaling of integrated circuits also depends on the development of new device fabrication processes and materials that allow the continued miniaturization of device features. As devices are scaled down, their gate lengths, gate oxide thicknesses, and channel lengths are reduced to the point where the channel conductance is no longer adequately modeled using conventional semiconductor physics.

1. List of Symbols

- Symbol: Parameter
- Symbol: Parameter

Charge on the electron:
- Symbol: Parameter

Effective charge:
- Symbol: Parameter

Absolute temperature:
- Symbol: Parameter

Drain current:
- Symbol: Parameter

Bolzmann's constant:
- Symbol: Parameter

Uniaxial strain constant:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Channel length:
- Symbol: Parameter

MOSFET channel length:
- Symbol: Parameter

Effective surface mobility:
- Symbol: Parameter

Source and drain doping levels:
- Symbol: Parameter

Channel doping level:
- Symbol: Parameter

Threshold voltage:
- Symbol: Parameter

Source and drain depletion layer widths:
- Symbol: Parameter

MOSFET channel width:
- Symbol: Parameter

Charge on the electron:
- Symbol: Parameter

Effective charge:
- Symbol: Parameter

Absolute temperature:
- Symbol: Parameter

Drain current:
- Symbol: Parameter

Bolzmann's constant:
- Symbol: Parameter

Uniaxial strain constant:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Channel length:
- Symbol: Parameter

MOSFET channel length:
- Symbol: Parameter

Effective surface mobility:
- Symbol: Parameter

Source and drain doping levels:
- Symbol: Parameter

Channel doping level:
- Symbol: Parameter

Threshold voltage:
- Symbol: Parameter

Source and drain depletion layer widths:
- Symbol: Parameter

MOSFET channel width:
- Symbol: Parameter

Charge on the electron:
- Symbol: Parameter

Effective charge:
- Symbol: Parameter

Absolute temperature:
- Symbol: Parameter

Drain current:
- Symbol: Parameter

Bolzmann's constant:
- Symbol: Parameter

Uniaxial strain constant:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Thermal expansion coefficient:
- Symbol: Parameter

Channel length:
- Symbol: Parameter

MOSFET channel length:
- Symbol: Parameter

Effective surface mobility:
- Symbol: Parameter

Source and drain doping levels:
- Symbol: Parameter

Channel doping level:
- Symbol: Parameter

Threshold voltage:
- Symbol: Parameter

Source and drain depletion layer widths:
- Symbol: Parameter

MOSFET channel width:
SCALING:
- Voltage: $V/\alpha$
- Oxide: $t_{ox}/\alpha$
- Wire width: $W/\alpha$
- Gate width: $L/\alpha$
- Diffusion: $x_d/\alpha$
- Substrate: $\alpha^*N_A$

RESULT:
- Higher density: $\sim\alpha^2$
- Higher speed: $\sim\alpha$
- Power/ckt: $\sim1/\alpha^2$

Power Density: $\sim$Constant
Moore’s Law

Introduction: Scaling-down

Transistors Per Die

- 1965 Actual Data
- MOS Arrays
- MOS Logic 1975 Actual Data
- 1975 Projection
- Memory
- Microprocessor

- Pentium® II
- Pentium® III
- Pentium® 4
- Itanium™

- 1G
- 2G
- 4G

Reasons of the shift

- Unacceptable leakage current
- Higher voltage implies higher performance

Result of the shift

Dramatically Increasing of power density

Introduction: Scaling-down
The Von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure (memory) to hold both instructions and data.

Characteristics of the Von Neumann's architecture:
- Memory;
- Control unit;
- Arithmetic Logic unit;
- Input/Output interface.

The disadvantage of Von Neumann architecture: shared memory for instructions and data with one data bus and one address bus between processor and memory. This shared memory is a bottleneck reducing the operation bandwidth.
Instruction = Low Efficiency

- Requiring a lot of preparations
- Accelerating all steps
- Accessing external memories
- Global connections

2+3=5

Introduction: Computation

Performance Wall
Memory Wall
Power Wall

Calculation
60 years ago, hardware was very expensive. Multiplexing resources is a must.
### Introduction: Economy

<table>
<thead>
<tr>
<th>Node</th>
<th>Estimated Cost</th>
<th>Flagship Tech Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>65nm</td>
<td>~2.5-3B $</td>
<td></td>
</tr>
<tr>
<td>45nm</td>
<td>~3.5-5B $</td>
<td></td>
</tr>
<tr>
<td>32nm</td>
<td>~5-7B $</td>
<td></td>
</tr>
<tr>
<td>22nm</td>
<td>~8-10B $</td>
<td></td>
</tr>
<tr>
<td>16nm</td>
<td>~12-15B $</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- NXP: Announced decision to not continue with Crolles TSMC ecosystem.
  - Will be fabless at 65nm and will eventually opt for 45nm.
- TI: Announced that process folding will be carried out primarily in conjunction with strategic partners.
  - Will increase mask runs of 32nm but will continue to have low or no internal capacity at 32nm.
- Infineon: Announced move to fabless model for logic products (excluding Qimonda).
  - Paradoxically, Infineon is using Chartered and UMC for wafer fab.
- LSI: Sells wafer fab, but has acquired partial ownership with Microsemi for wafer fab through Agere Systems merger.
  - Will likely continue to be part of the fab, but will be a fabless at 45nm.
- TSMC: Continues to be a part of the fab, but is acquiring Toshiba and NEC Electronics as well.
Beyond 22nm, cost-reduction is no longer a major object. But increasing performance is still a main task.

### Lower Utilization

<table>
<thead>
<tr>
<th>Technology node (µm)</th>
<th>K gates per sq. mm</th>
<th>Utilization (%)</th>
<th>Gate count (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Potential</td>
<td>Actual</td>
<td>Average</td>
</tr>
<tr>
<td>0.25</td>
<td>112</td>
<td>92 to 97</td>
<td>94.2</td>
</tr>
<tr>
<td>0.18</td>
<td>164</td>
<td>133 to 140</td>
<td>136.7</td>
</tr>
<tr>
<td>0.13</td>
<td>248</td>
<td>199 to 209</td>
<td>204.0</td>
</tr>
<tr>
<td>0.090</td>
<td>443</td>
<td>344 to 361</td>
<td>352.2</td>
</tr>
<tr>
<td>0.065</td>
<td>694</td>
<td>500 to 545</td>
<td>522.9</td>
</tr>
<tr>
<td>0.045</td>
<td>1,179</td>
<td>778 to 871</td>
<td>824.7</td>
</tr>
<tr>
<td>0.032</td>
<td>1,723</td>
<td>1,023 to 1,173</td>
<td>1,098.4</td>
</tr>
<tr>
<td>0.022</td>
<td>2,726</td>
<td>1,450 to 1,682</td>
<td>1,566.1</td>
</tr>
</tbody>
</table>

### Gross Profit Margin

<table>
<thead>
<tr>
<th>%</th>
<th>0.13µm</th>
<th>90nm</th>
<th>65nm</th>
<th>45nm</th>
<th>32nm</th>
<th>22nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross profit margin</td>
<td>46.0</td>
<td>48.0</td>
<td>50.0</td>
<td>52.0</td>
<td>55.0</td>
<td>57.0</td>
</tr>
<tr>
<td>• Product development R&amp;D</td>
<td>14.0</td>
<td>17.0</td>
<td>20.0</td>
<td>23.0</td>
<td>27.0</td>
<td>30.0</td>
</tr>
<tr>
<td>• SG&amp;A</td>
<td>12.0</td>
<td>11.0</td>
<td>10.0</td>
<td>9.0</td>
<td>8.0</td>
<td>7.0</td>
</tr>
<tr>
<td>• Operating income</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Cost of goods sold</td>
<td>54.0</td>
<td>52.0</td>
<td>50.0</td>
<td>48.0</td>
<td>45.0</td>
<td>43.0</td>
</tr>
<tr>
<td>• Wafer costs</td>
<td>35.0</td>
<td>35.0</td>
<td>35.0</td>
<td>34.0</td>
<td>33.0</td>
<td>26.0</td>
</tr>
<tr>
<td>• Packaging costs</td>
<td>10.0</td>
<td>9.0</td>
<td>8.0</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
</tr>
<tr>
<td>• Testing costs</td>
<td>9.0</td>
<td>8.0</td>
<td>7.0</td>
<td>6.0</td>
<td>6.0</td>
<td>7.0</td>
</tr>
<tr>
<td>Revenues</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Only a few high-end chip makers today can even afford the exorbitant cost of NEXT-GENERATION RESEARCH AND DESIGN, much less the fabs to build them.

将来只有少数高端芯片设计公司可以负担昂贵的研发费用，而更少的公司有能力制造新一代的产品。

R. Colin Johnson,
“IBM Fellow: Moore’s Law Defunct,” EE Times, 4/07/09

Wally Rhines, Chairman & CEO, Mentor Graphics, August 2010
Introduction: How Far We Can Go?

Performance, Cost and Power

- Performance is a lasting theme
- Reducing cost while keeping performance
- State-of-arts Design skills
- Trade-off Compromise

Reducing power consumption While keeping performance and cost
High-Performance, Low Power, Low Cost

Introduction: How Far We Can Go?
Summary

- Moore’s Law meets a serious challenge of power wall.
- Traditional Von Neumann architecture meets the difficulty of low efficiency.
- Increasing cost will stop most investment.
- Performance is an never-ending theme.
- Over 16nm node, technology will go on.
- On product configuration, great changes will be taken place.
- Few companies and few general, platform based products will dominate market.
Outlines

- Introduction
- Von Neumann Architecture
- Reconfigurable Computing
- Major Challenges
- Conclusions
The Von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure (memory) to hold both instructions and data.

Characteristics of the Von Neumann’s architecture:
- a) memory;
- b) control unit;
- c) arithmetic logic unit;
- d) input / output interface.

The disadvantage of Von Neumann architecture: shared memory for instructions and data with one data bus and one address bus between processor and memory. Instructions and data have to be fetched in sequential order (known as the Von Neumann Bottleneck), limiting the operation bandwidth.
Architecture Evolution

Von Neumann Architecture

Memory

Control Unit
Arithmetic Logic Unit
Input Output

I-Memory

Control Unit
Arithmetic Logic Unit
I/O

D-Memory

Control Unit
Arithmetic Logic Unit
I/O

Memory

Control Unit
Arithmetic Logic Unit
Input

Memory

Control Unit
Arithmetic Logic Unit
Input

Memory

Controller
Datapath
Input

Memory

Controller
Datapath
Input

Memory

Controller
Datapath
Input

Memory

Controller
Datapath
Input
This universal architecture leads to different structures for ASIC and general purpose processor.
ASP: Datapath and Controller

Datapath consists in resources, memories and interconnections.
Datapath performs functional calculation according to inputs.
Datapath operates according to control-vectors from controller.
Datapath is not directly controlled by system clock.
Datapath provides necessary test-data to controller.

Datapath consists in
- Memory
- Memories
- Interconnections

Datapath performs functional calculation according to inputs.
Datapath operates according to control-vectors from controller.
Datapath is not directly controlled by system clock.
Datapath provides necessary test-data to controller.

Controller consists in state generator and control-vector generator.
Controller generates control-vectors according to test-data and data-flow.
Controller operates according to system clock.

ASP: Application Specific Processor → AISC
Any digital system can be composed of Datapath and Controller.
ASIC: Uniform Architecture

\[ z = x + y \]

Operation: Addition

Variable: \( z \)

Register

ALU

Hardware Descriptions

BEGIN
END

END LOOP;

\( r_2 = r_3 / 2 \);

\( r_3 = r_3 + r_4 \);

\( r_4 = r_1 / r_3 \);

FOR \( i \) IN \( [1..n] \) LOOP:

\( r_3 = r_3 - 0.2 \);

\( r_3 = r_3 + 0.1 \);

\( r_3 = r_3 - 0.4 \);

BEGIN

SQR(r) 16
RTL Structure

ASIC Design: High-Level Synthesis
1-Dimension Expansion

- Parallelizing calculations implied by data dependency.
- Interconnections only exist according to system function.
SQR(r1) is
BEGIN
  r3 := r1 \* 4;
  r3 := r3 + 1;
  r3 := r3 \* 0.2;
  FOR i IN INTEGER RANGE 0 TO 3 LOOP
    r4 := r1 / r3;
    r3 := r3 + r4;
    r3 := r3 / 2;
  END LOOP;
END;

- Application specified architecture
- Scheduling/allocation during design
- Depends on hardware description
- Components do not fully connected
- Control-Codes cannot be changed
- State-Machine is not programmed

Hardware Descriptions

Operation Scheduling
Register Optimization
Resource Allocation
Interconnection Generation
Control-Codes Generation
State Generation
FSM Design
Datapath consists in resources, memories and interconnections.

Datapath performs functional calculation according to inputs.

Datapath operates according to control-vectors from controller.

Datapath is not directly controlled by system clock.

Datapath provides necessary test-data to controller.

Controller consists in state generator and control-vector generator.

Controller generates control-vectors according to test-data and program.

Controller operates according to system clock.

GPP: General Purpose Processor $\rightarrow$ MPU

Any processor can also be composed of Datapath and Controller.
SQR(r1) is
BEGIN
  r3 := r1 * 4;
  r3 := r3 + 1;
  r3 := r3 * 0.2;
  FOR i IN INTEGER RANGE 0 TO 3 LOOP
    r4 := r1 / r3;
    r3 := r3 + r4;
    r3 := r3 / 2;
  END LOOP;
END;

- General purpose architecture
- Scheduling/allocation during compile
- Interconnection, control-codes and state are generated during compile
- Components are fully connected
- Control-Codes can be changed
- State-Machine is programmable
Von Neumann architecture describes a universal computing machine, but with low efficiency.

A universal architecture for digital system can be generated from Von Neumann architecture that consists in datapath and controller.

The methodology of High-Level Synthesis is a way to design ASIC, which aims at hardware design.

However, with a fully configurable hardware, the methodology of High-Level Synthesis may be done in software and implement a general purpose processor.
Von Neumann Architecture
Reconfigurable Computing
Major Challenges
Conclusions
## Comparisons

<table>
<thead>
<tr>
<th>Products</th>
<th>Performance</th>
<th>Cost</th>
<th>Power</th>
<th>Flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIC</td>
<td>Very High</td>
<td>Very Low</td>
<td>Very Low</td>
<td>Poor</td>
</tr>
<tr>
<td>FPGA</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Good</td>
</tr>
<tr>
<td>CPU</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Very Good</td>
</tr>
<tr>
<td>RCP</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Very Good</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Products</th>
<th>Performance</th>
<th>MOPS/mm²</th>
<th>nJ/Operation</th>
<th>Programmability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded RISC</td>
<td>Average</td>
<td>&lt;100</td>
<td>~1</td>
<td>Very Good</td>
</tr>
<tr>
<td>DSP</td>
<td>Average</td>
<td>&lt;100</td>
<td>~1</td>
<td>Very Good</td>
</tr>
<tr>
<td>Multi-Core RISC</td>
<td>High</td>
<td>&lt;100</td>
<td>~1</td>
<td>Good</td>
</tr>
<tr>
<td>Multi-Core DSP</td>
<td>High</td>
<td>&lt;100</td>
<td>~1</td>
<td>Good</td>
</tr>
<tr>
<td>ASIC</td>
<td>Very High</td>
<td>&gt;1000</td>
<td>~0.01</td>
<td>Poor</td>
</tr>
<tr>
<td>RCP</td>
<td>High</td>
<td>~1000</td>
<td>~0.03</td>
<td>Good</td>
</tr>
</tbody>
</table>

ASIC: Application Specific Integrated Circuit  
FPGA: Field Programmable Gate Array  
CPU: Central Processing Unit  
RCP: Re-Configurable Processor
Uniform Architecture

SQR(r1) is
BEGIN
  r3 := r1 * 4;
  r3 := r3 + 1;
  r3 := r3 * 0.2;
FOR I IN INTEGER RANGE 0 TO 3 LOOP
  r4 := r1 / r3;
  r3 := r3 + r4;
  r3 := r3 / 2;
END LOOP;
END;

Program

\[ z = x + y \]

Datapath

Memory

Controller

Input
Test Data
Control Vectors
Clock
Program

Output
IF (condition = 1) THEN
  r1 = b * b;
  r2 = a * c;
  r2 = r2 * 4;
  r1 = r1 - r2;
  IF (r1 < 0) THEN
    state = ‘1’;
  ELSE
    state = ‘0’;
    r3 = r1 * 4;
    r3 = r3 + 1;
    r3 = r3 * 0.2;
    FOR i = 1 to 3 LOOP
      r4 = r1 / r3;
      r3 = r3 + r4;
      r3 = r3 / 2;
    END LOOP;
    r1 = 0 - b;
    r2 = a + a;
    r4 = r1 + r3;
    x1 = r4 / r2;
    r5 = r1 - r3;
    x2 = r5 / r2;
  END IF;
END IF;

IF (condition = 1) THEN
  COMPTATION 1;
  IF (r1 < 0) THEN
    COMPTATION 2;
  ELSE
    COMPTATION 3;
    COMPTATION 4;
    END LOOP;
    COMPTATION 5;
  END IF;
END IF;

COMPTATION 1
  r1 = b * b;
  r2 = a * c;
  r2 = r2 * 4;
  r1 = r1 - r2;

COMPTATION 2
  state = ‘1’;

COMPTATION 3
  state = ‘0’;
  r3 = r1 * 4;
  r3 = r3 + 1;
  r3 = r3 * 0.2;

COMPTATION 4
  r4 = r1 / r3;
  r3 = r3 + r4;
  r3 = r3 / 2;

COMPTATION 5
  r1 = 0 - b;
  r2 = a + a;
  r4 = r1 + r3;
  x1 = r4 / r2;
  r5 = r1 - r3;
  x2 = r5 / r2;
1-Dimension Expansion

1-Dimension Data-path

RTL Architecture
Reconfigurable Computing: Datapath

Two-dimension Expansion

2-Dimension Data-path

\( x \)

\( y \)
3-Dimension Data-path

Reconfigurable Computing: Datapath

Three-dimension Expansion

x

y

z
\[
e = a + b; \\
f = c \times d; \\
h = e - f; \\
h = e + f; \\
i = e / f; \\
o_1 = g \times h; \\
o_2 = h - i;
\]
e = a + b;
f = c * d;
h = e - f;
h = e + f;
i = e / f;
o1 = g * h;
o2 = h - l;
Interconnections among operation 1, operation 2 and operations 3, operation 4, operation 5 are configured according to the data dependency.
Processors, memories and interconnections that are not used will be power gated in order to decrease power consumption.
Datapath: Control Intensive

\[ v_1 = a + b; \]

Case (t)
1: \( v_1 = v_1 \times t; \)
2: \( v_2 = v_1 - t; \)
3: \( v_3 = v_1 + t; \)
End case;
\[ o_1 = v_1 + v_2; \]
\[ o_2 = v_2 - v_3; \]
v1 = a + b;
Case (t)
1: v1 = v1 * t;
2: v2 = v1 - t;
3: v3 = v1 + t;
End case;
o1 = v1 + v2;
o2 = v2 - v3;
Connections Configuration

Two ways to determine the final result:
1: Using configuration data to fix the wiring;
2: Calculating all the paths and then to decide which one will be taken.

Interconnections among operation 1, operation 2 and operations 3, operation 4, operation 5 are selective depending on the variable t.
Processors, memories and interconnections that are not used will be power gated in order to decrease power consumption.
MLU: Multi-Function Unit

Reconfigurable Computing: Datapath

Context Interface  IN Interface  OUT Interface

INPUT BUFFER  OUTPUT BUFFER  INPUT BUFFER

Context  IRQs  Status Controller

Context Interface

Reconfigurable Computing: Datapath

Datapath

INPUT BUFFER  OUTPUT BUFFER

Temp data  Results from previous line

MUX A  Sel A

MUX B  Sel B

ALU  Op_code

MUX C  Sel_C

Temp reg  Result reg

Context

Data

X-Bus

Temp data  Results from previous line

MUX A

MUX B

ALU

MUX C

Temp reg  Result reg

Context

Data

Y-Bus
IF (condition = 1) THEN
    COMPUTATION 1;
IF (r1 < 0) THEN
    COMPUTATION 2;
ELSE
    COMPUTATION 3;
END IF;
FOR i = 1 to 3 LOOP
    COMPUTATION 4;
END LOOP;
COMPUTATION 5;
END IF;
RB-PFSM: 1. Read current state from State-Memory
2. Decide next state and generate address according to test data, that address the Data-Memory and Configuring-Context
3. Output the data and configuring-context
MB-PFSM: 1. Read current state from State/Address memory
2. Decide next state and generate address according to test data, that address the Data-Memory and Configuring-Context
3. Output the data and configuring-context
Reconfigurable Computing: Controller

Control-Flow & Configuring Context

Controller

RISC
State Control
State/Address Memory

Global-Data Memory
Configuring-Context
Register Enable/Disable
ALU-Context
Connection-Context
Power-Context

Datapath

Register
ALU
Connection Config.
Power Config.

Global Data
Register En/Disable
ALU Config.

Test Data
Only one row works each time when datapath performs calculation. Thus, the power consumption of datapath is greatly reduced.
Multi-Task Pipeline Execution

Tasks are independent

Number of tasks is limited by the size of datapath
### Compiler

#### High-Level Programming Language (Such as: ANSI C)

```c
Int main(void) {
    ......
    func(..., ...)
    ......
    dct(..., ...,)
    ......
}
```

```c
func(..., ...)
{
    ......
}
```

```c
dct(..., ...,)
{
    ......
}
```

### Compiler

#### Syntax Check
- Code Profiling
- Code Transformation
- Code Optimization
- Data-Flow Generation
- Task Partitioning
- Task Scheduling
- Allocation
- Connection Scheme
- Mappings
- Evaluations
- Context Generation

```
if ((a+b) > 1) {
    ...
}
```

```
c = a + b
if (c > 1) {
    ...
}
```

```
......
{... r3 := r1 * 4;
 r3 := r3 + 1;
 r3 := r3 * 0.2
 for i = 0 to 3 {
     r4 := r1 / r3;
     r3 := r3 + r4;
     r3 := r3 / 2;
 }......
```
Compiler: Special Features

High-Level Programming Language
(Such as: ANSI C)

```c
int main(void)
{
    ......
    func(..., ...)
    ......
    dct(..., ...,)
    ......
    }
}
```

Reconfigurable Computing

Compiler

- Syntax Check
- Code Profiling
- Code Transformation
- Code Optimization
- Data-Flow Generation
- Task Partitioning
- Task Scheduling
- Allocation
- Connection Scheme
- Mappings
- Evaluations
- Context Generation

- Code modification
- Adaptation
- Parallelization
- Optimization
- Efficiency
- Parallelization
- Sharing
- Localization
- Simplicity
- Short delay
- Power saving
- Utilization cost
- Power

Reconfigurable Processor
Summary

- Unified computing mode.
- Multi-dimension data-path.
- Multi-function ALU.
- Operations mapping.
- Localization of memories.
- Regularity of connections.
- Power gating technology.
- Programmable Finite-State Controller.
- New compiler.
Basic Requirement

- No Instruction
- C Language Programming
- Dynamic Reconfigurable
- Localizing Communications
- Localizing Memories
- Scalable & Extendable

- Existing Software Porting
- No IPR Issues
- Reducing Global Wires
- Hardware Flexibility
- Reducing Memory Wall
- Improving Flexibility
Programming Language

High-Level Programming Language (Such as: ANSI C)

```c
int main(void)
{
    ....
    func(..., ...)
    ....
    dct(..., ...,)
    ....
    ....
}
```

```c
func(..., ...)
{
    ....
}
```

```c
dct(..., ...)
{
    ....
}
```

Challenges

- Available software
- C language
- Pointers
- Loops
- Recursive call
- Vector
- matrix
- ....

Compiler

- Syntax Check
- Code Profiling
- Code Transformation
- Code Optimization
- Data-Flow Generation
- Task Partitioning
- Task Scheduling
- Allocation
- Connection Scheme
- Mappings
- Evaluations
- Context Generation
Dimension Limited: Task Mapping

Task Partitioning
Task Flow Generation
Task Dependency

Task Scheduling
Datapath Allocation
Mapping Scheme
**Partitioning and Deadlock**

Task Graph Partitioning (No Deadlock)

- **Requirements**
  - Less interconnections (communication cost) between sub-parts;
  - Avoid **deadlock**.

Task Graph Partitioning (Deadlock)
Global Data Passing

Task Graph Partitioning

Global Data Passing

Major Challenges

Global Data Memory

Cache
Operating System

Major Challenges:
- Multi-Task management
- Resource management
- Time sharing and resource sharing

PROGRAM A
```c
int main(void)
{
    ....
    func(..., ...)
    ....
    dct(..., ...,)
    ....
    ....
}
```

PROGRAM B
```c
int main(void)
{
    ....
    func(..., ...)
    ....
    dct(..., ...,)
    ....
    ....
    ....
}
```

PROGRAM C
```c
int main(void)
{
    ....
    func(..., ...)
    ....
    dct(..., ...,)
    ....
    ....
}
```

- Multi-Controller
- Dynamic Resource allocation
- Queue-up
Regarding available software, C language must be considered as programming language. Many problems, such as “pointer”, will be studied. Dimension limited, task mapping iterates. Partitioning and mapping algorithms need to be found. Deadlock should be avoided. Global memory is necessary and data passing scheme is a important issue. New compiler instead of traditional one. New operation system should be developed so that it supports massive parallel execution.
Outlines

- Introduction
- Von Neumann Architecture
- Reconfigurable Computing
- Major Challenges
- Conclusions
Conclusions

1. Technology will go on beyond 16nm, but only few companies and few general products will dominate market.
2. From single-core to many-core, from simple FPGA to processing heterogeneous array, reconfigurable computing will be the next trend.
3. Rich computing resources, localization of memories, regularity connections, general task flow mapping and great power efficiency, etc., all of these make up a reconfigurable, programmable processing array.
4. Problems are still tough, such as programming language, partitioning and mapping, extra memories, new operating system and so on.
5. In the future design, compiler is a key.
Prediction is very difficult, especially if it’s about the future.

-Niels Bohr

尼尔斯·波尔（1885-1962），丹麦物理学家。因原子结构和原子辐射的研究，获得1922年的诺贝尔物理学奖。
Anyone who says businessmen deal in facts—not fiction—has never read old five-year projections...”

- Malcolm Forbes

假如有人说生意的经营是基于事实而非空想，那他一定没有看过那些过期的五年经营规划书。

Wally Rhines, Chairman & CEO, Mentor Graphics, August 2010
Thank You for Your Attention