清华大学移动计算研究中心

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

Scaling-down

P4004：1971年世界上第一个微处理器在英特尔公司诞生

Von Neumann

Von Neumann Architecture

Semiconductor

Computer

Moore's Law

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

units.

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.

Semiconductor and has been director of the research and development laboratories since 1959.

in equipment today.

Electronics, Volume 38, Number 8, April 19, 1965

Robert Dennard and Dennard's Law

Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions

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

Classic Paper

This paper considers the design, fabrication, and characterization of very small MOSFET switching devices suitable for digital integrated circuits using dimensions of the order of 1 μ. Scaling relationships are presented which show how a conventional MOSFET can be reduced in size. An improved small device structure is presented that uses ion implantation to provide shallow source and drain regions and a nonuniform substrate doping profile. One-dimensional models are used to predict the substrate doping profile and the corresponding threshold voltage versus source voltage characteristic. A two dimensional current transport model is used to predict the relative degree of short channel effects for different device parameter combinations. Polysilicon gate MOSFET's with channel lengths as short as 0.5 μ were fabricated, and the device characteristics measured and compared with predicted values. The performance improvement expected from using these very small devices in highly miniaturized integrated circuits is projected.

I. List of Symbols

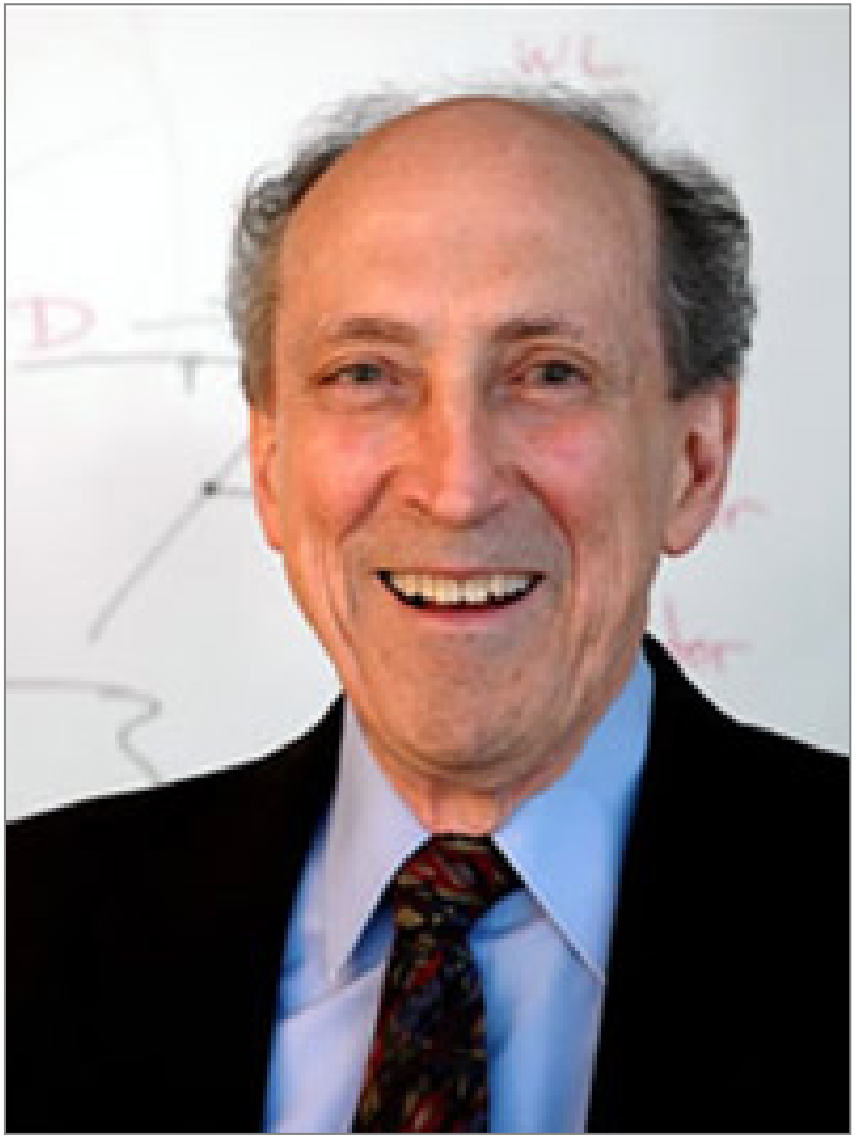
α	Inverse semilogarithmic slope of subthreshold characteristic.
D	Width of idealized step function profile for channel implant.
ΔW_f	Work function difference between gate and substrate.
ϵ_{Si}; ϵ_{ox}	Dielectric constants for silicon and silicon dioxide.
I_d	Drain current.
k	Boltzmann's constant.
K	Unitless scaling constant.
L	MOSFET channel length.
μ_{eff}	Effective surface mobility.
n_i	Intrinsic carrier concentration.
N_a	Substrate acceptor concentration.
Ψ_s	Band bending in silicon at the onset of strong inversion for zero substrate voltage.
Ψ_b	Built-in junction potential.
q	Charge on the electron.
Q_{ss}	Effective oxide charge.
t_{ox}	Gate oxide thickness.
T	Absolute temperature.
V_d, V_s, V_g, V_{sub}	Drain, source, gate and substrate voltages.
V_{ds}	Drain voltage relative to source.
V_{s-sub}	Source voltage relative to sustrate.
V_t	Gate threshold voltage.
w_s, w_d	Source and drain depletion layer widths.
W	MOSFET channel width.

II. Introduction

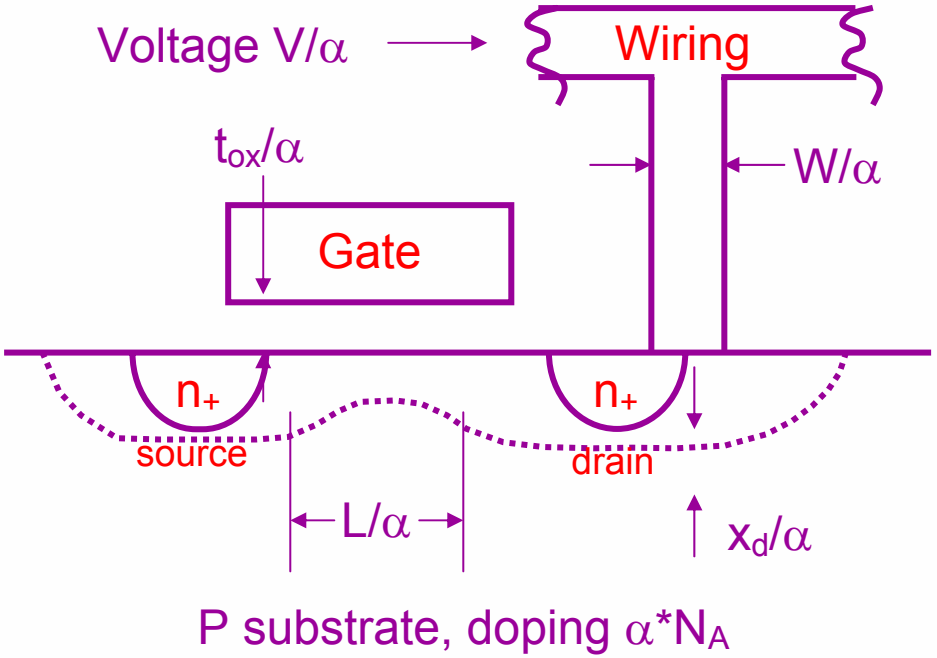
New high resolution lithographic techniques for forming semiconductor integrated circuit patterns offer a decrease in linewidth of five to ten times over the optical contact masking approach which is commonly used in the semiconductor industry today. Of the new techniques, electron beam pattern writing has been widely used for experimental device fabrication [1]–[4] while X-ray lithography [5] and optical projection printing [6] have also exhibited high-resolution capability. Full realization of the benefits of these new high-resolution lithographic techniques requires the development of new device designs, technologies, and structures which can be optimized for very small dimensions.

This paper concerns the design, fabrication, and characterization of very small MOSFET switching devices suitable for digital integrated circuits using dimensions of the order of 1 μ. It is known that reducing the source-to-drain spacing (i.e., the channel length) of an FET leads to undesirable changes in the device characteristics. These changes become significant when the depletion regions surrounding the source and drain extend over a large portion of the region in the silicon substrate under the gate electrode. For switching applications, the most undesirable "short-channel" effect is a reduction in the gate threshold voltage at which the device turns on, which is aggravated

This paper is reprinted from IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. SC-9, no. 5, pp. 256–268, October 1974.
Publisher Item Identifier S 0018-9219(99)02196-9.



Scaling-Down vs. Power Issue



SCALING:

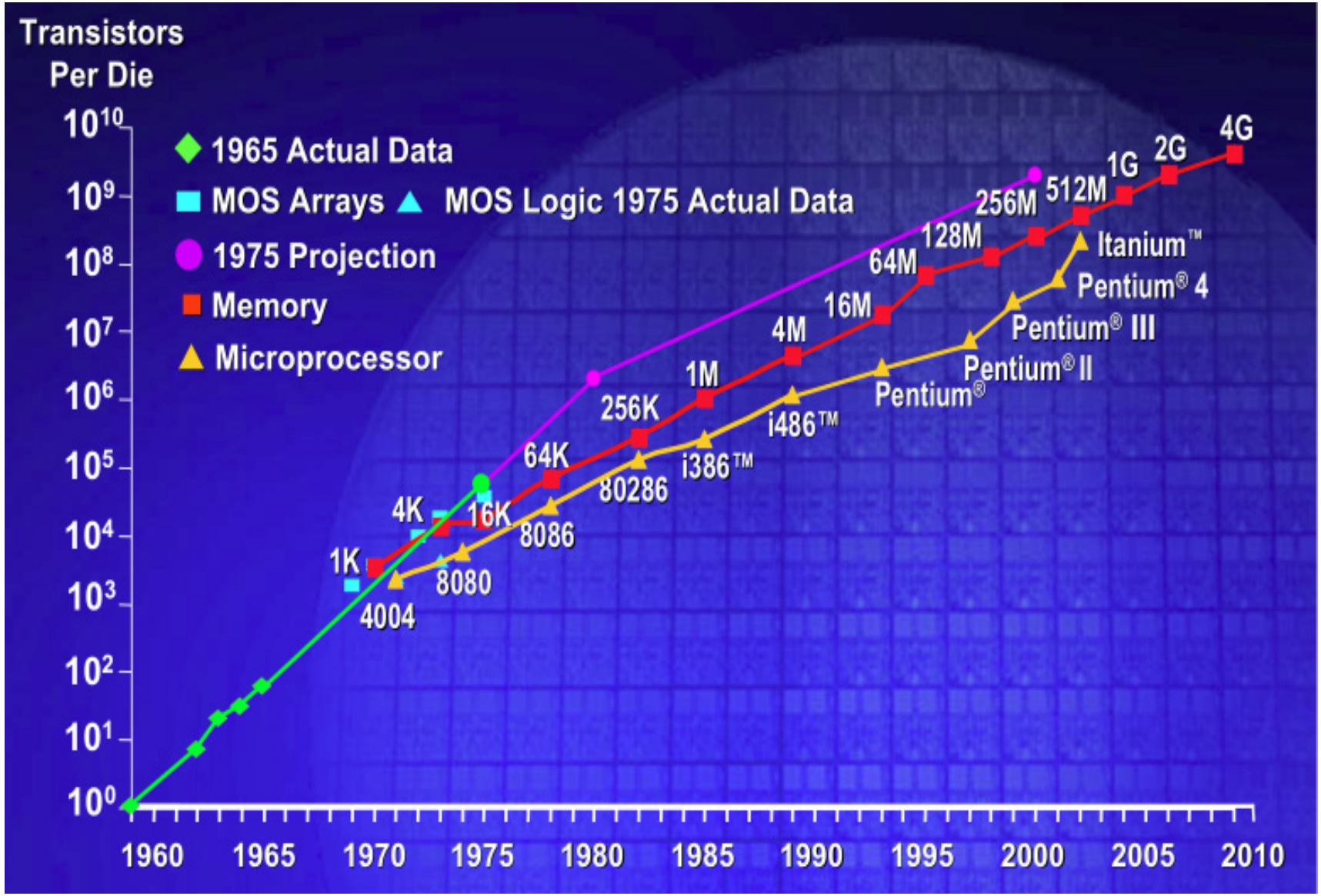
Voltage:	V/α
Oxide:	t_{ox}/α
Wire width:	W/α
Gate width:	L/α
Diffusion:	x_d/α
Substrate:	$\alpha * N_A$

RESULT:

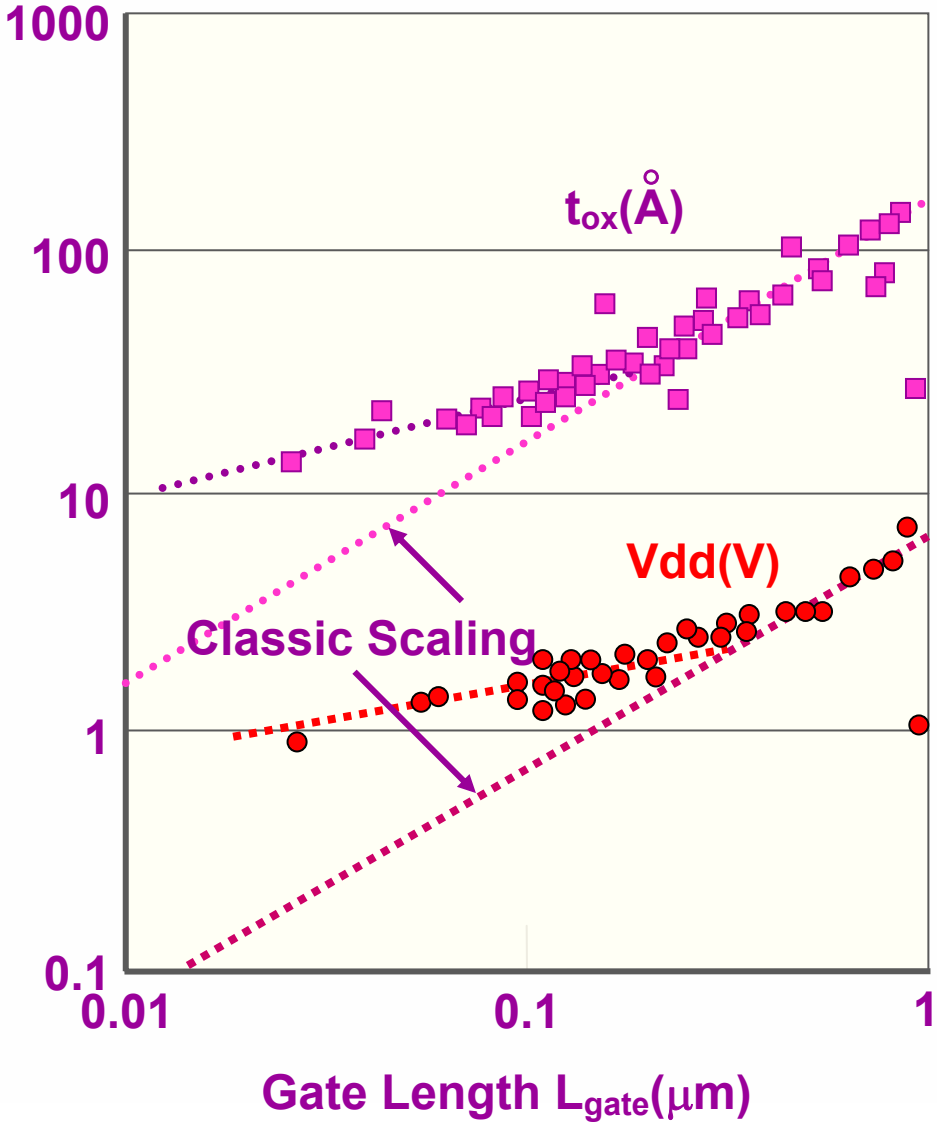
Higher density:	$\sim\alpha^2$
Higher speed:	$\sim\alpha$
Power/ckt:	$\sim 1/\alpha^2$

Power Density: ~Constant

Moore's Law


Transistors
Per Die
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
1965 Actual Data
MOS Arrays
MOS Logic 1975 Actual Data
1975 Projection
Memory
Microprocessor
1K
4K
16K
64K
256K
1M
4M
16M
64M
128M
256M
512M
1G
2G
4G
4004
8080
8086
80286
i386™
i486™
Pentium®
Pentium® II
Pentium® III
Pentium® 4
Itanium™
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010

Power Density ≠ Constant



Reasons of the shift

- Unacceptable leakage current
- Higher voltage implies higher performance

Result of the shift

Dramatically Increasing of power density

Von Neumann Architecture

The **Von Neumann architecture** is a design m… **store**… **digit**… **r** t… ent… u… at… ct… t… nst… dat…

Pipeline

Multi-threads

Out-of-order Execution

Characteristics of the Von Neumann's architecture:

…ry;

…un…

…tic…

…out…

Harvard

VLIW

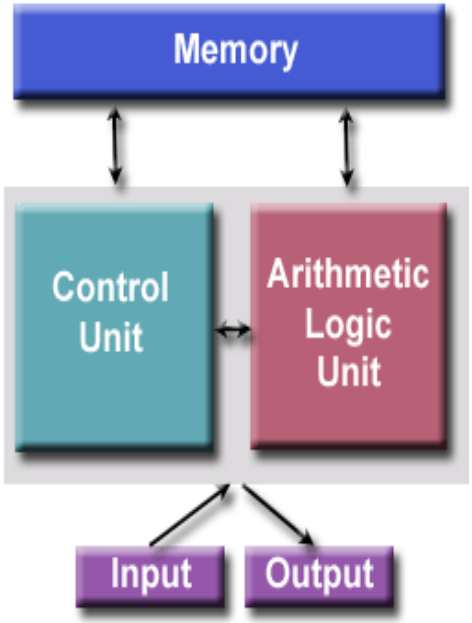
Multi-Issues

The disadvantage of Von Neumann architecture: shared memory for instructions and… …ith one… …s and o… …ess h… …en… an… …nd… …o… …ord… a… N… …ottlen… …ng th… …n **bandwidth**.

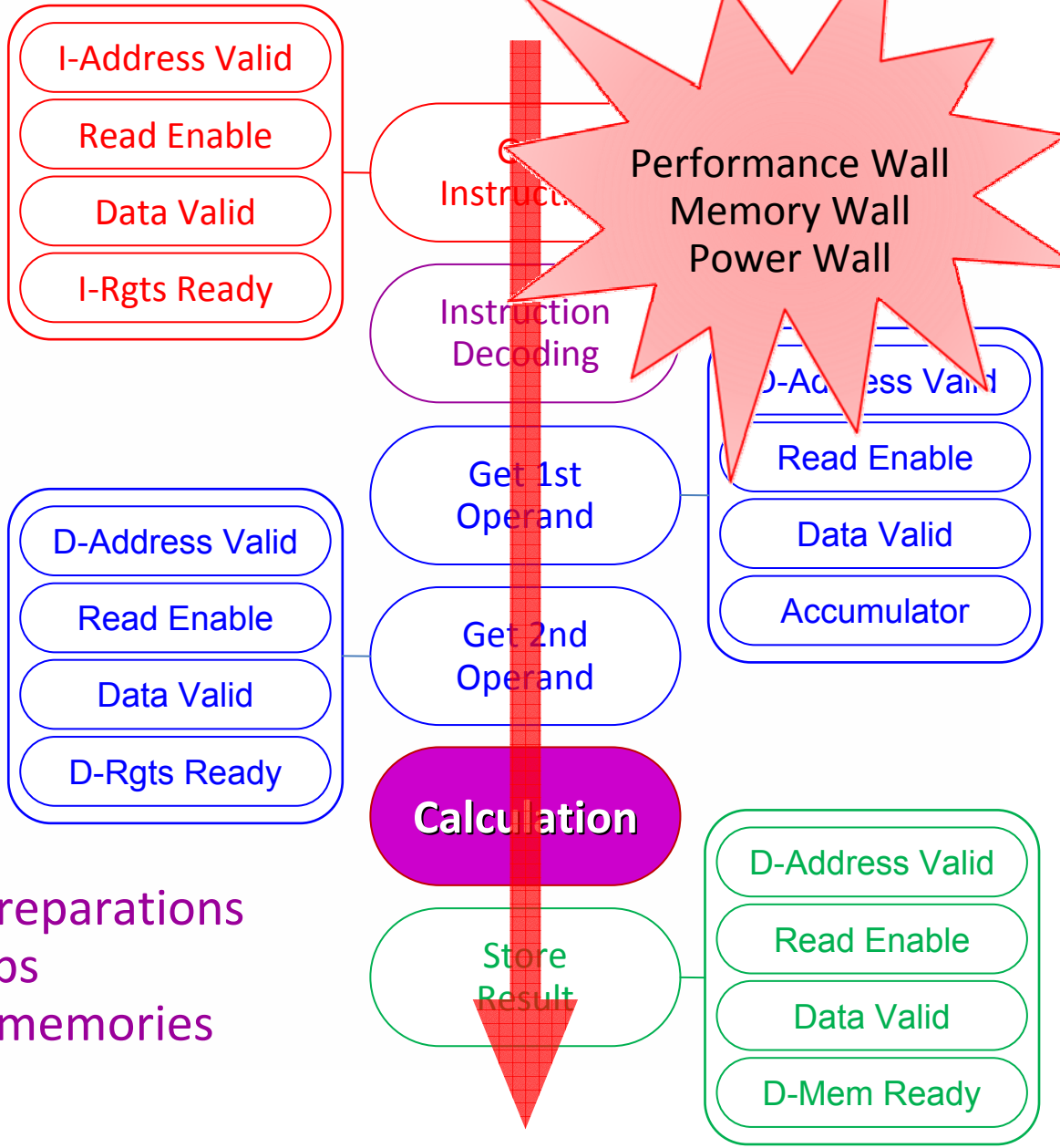
Dual-Cores

Multi-Cores

Many-Cores

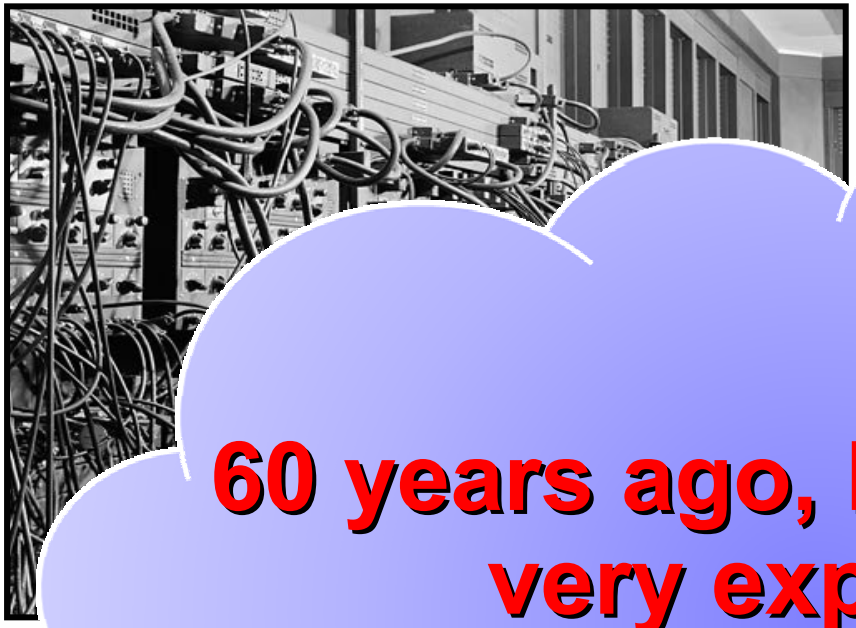


Instruction = Low Efficiency

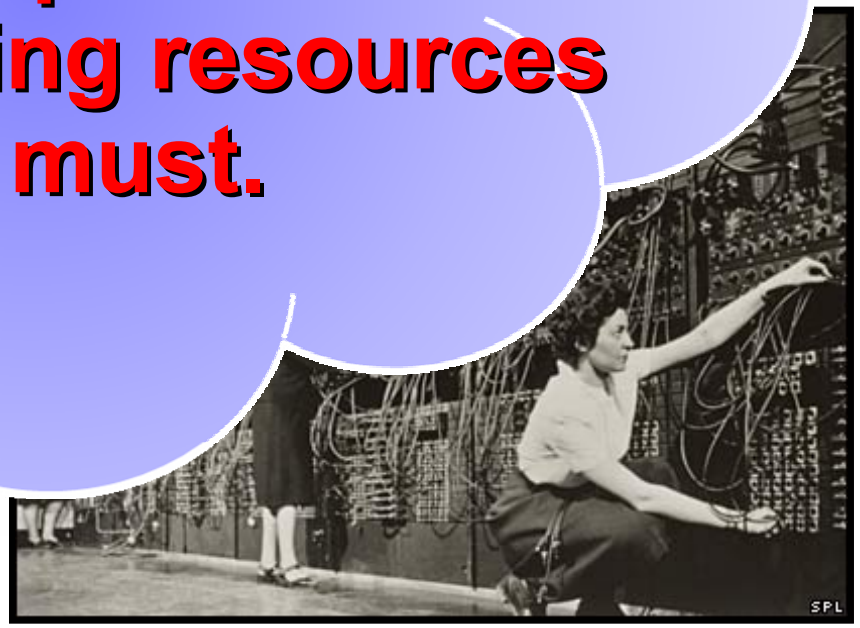


- Requiring a lot of preparations
- Accelerating all steps
- Accessing external memories
- Global connections

Hardware = Expensive



60 years ago, hardware was very expensive. Multiplexing resources is a must.



Huge Investment Stops Investor

	needs
NXP	Announced decision to not continue with Crolles … ded to join TSMC ecosystem Will be fabless at 65nm and … ally
TI	Announced that process … t at 32nm will be carried out primarily in conjunction with strateg… Will incr… …und… …ive low or no internal capacity at 32nm
Infineon	Anno… fabl… …ogic products (excluding Qimonda) Par… …d is using Chartered and UMC for wa…
LSI	So… …fab, but has acquired partial ownersh… with … …re through Agere Systems merger …ll like…
	…s to be fabless at 45…
	…tinues to be part of I…
	…Toshiba and NEC Electronics as well

16nm
～12-15B $

22nm
～8-10B $

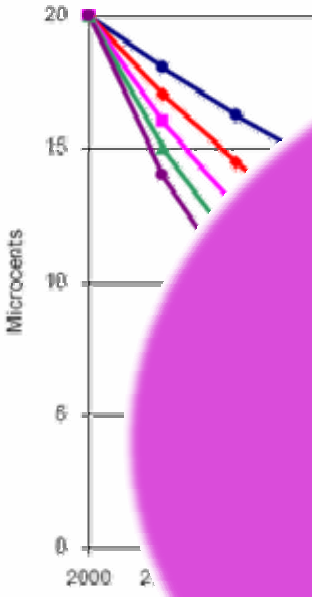
32nm
～5-7B $

45nm
～3.5-5B $

65nm
～2.5-3B $

Scaling-down ≠ Cost-down

Cost Per Transistor Red...



(US Dollar)

...nm	2,450
	2,810
	...06
	...

...2012 Perspective)

	...2nm	22nm
Cost per gate (mic...	6.89	6.66
• Percent decline (%)	(10.1)	(3.3)

Cost Per Gate Trends

Beyond 22nm, cost-reduction is no longer a major object. But increasing performance is still a main task.

Source：Industry Restructuring, IBS Report, 2007

Lower Utilization

Technology node (μm)	K gates per sq. mm			Utilization (%)	Gate count (M)		
	Potential	Actual	Average		5mm x 5mm	10mm x 10mm	20mm x 20mm
0.25	112	92 to 97	94.2	81.8 to 86.4	2.3 to 2.4	9.2 to 9.7	36.6 to 38.7
0.18	164	133 to 140	136.7	81.1 to 85.6	3.3 to 3.5	13.3 to 14.0	53.2 to 56.2
0.13	248	199 to 209	204.0	80.2 to 84.3	5.0 to 5.2	19.9 to 20.9	79.6 to 83.6
0.090	443	344 to 361	352.2	77.6 to 81.4	8.6 to 9.0	34.4 to 36.1	137.5 to 144.2
0.065	694	500 to 545	522.9	72.1 to 78.6	12.5 to 13.6	50.0 to 54.5	200.1 to 218.2
0.045	1,179	778 to 871	824.7	66.0 to 73.9	19.5 to 21.8	77.8 to 87.1	311.3 to 348.5
0.032	1,723	1,023 to 1,173	1,098.4	59.4 to 68.1	25.6 to 29.3	102.3 to 117.3	409.4 to 469.3
0.022	2,726	1,450 to 1,682	1,566.1	53.2 to 61.7	36.3 to 42.0	145.0 to 168.2	580.1 to 672.8

%	0.13μm	90nm	65nm	45nm	32nm	22nm
Gross profit margin	46.0	48.0	50.0	52.0	55.0	57.0
• Product development R&D	14.0	17.0	20.0	23.0	27.0	30.0
• SG&A	12.0	11.0	10.0	9.0	8.0	7.0
• Operating income	**20.0**	**20.0**	**20.0**	**20.0**	**20.0**	**20**
Cost of goods sold	54.0	52.0	50.0	48.0	45.0	43.
• Wafer costs	35.0	35.0	35.0	34.0	30.0	26.0
• Packaging costs	10.0	9.0	8.0	8.0	9.0	10.0
• Testing costs	9.0	8.0	7.0	6.0	6.0	7.0
Revenues	100.0	100.0	100.0	100.0	100.0	100.0

On 16nm node, the overall cost to develop a chip will be 150M-200M USD

Source：Industry Restructuring, IBS Report, 2007

Integrated Circuit: Game of Richman

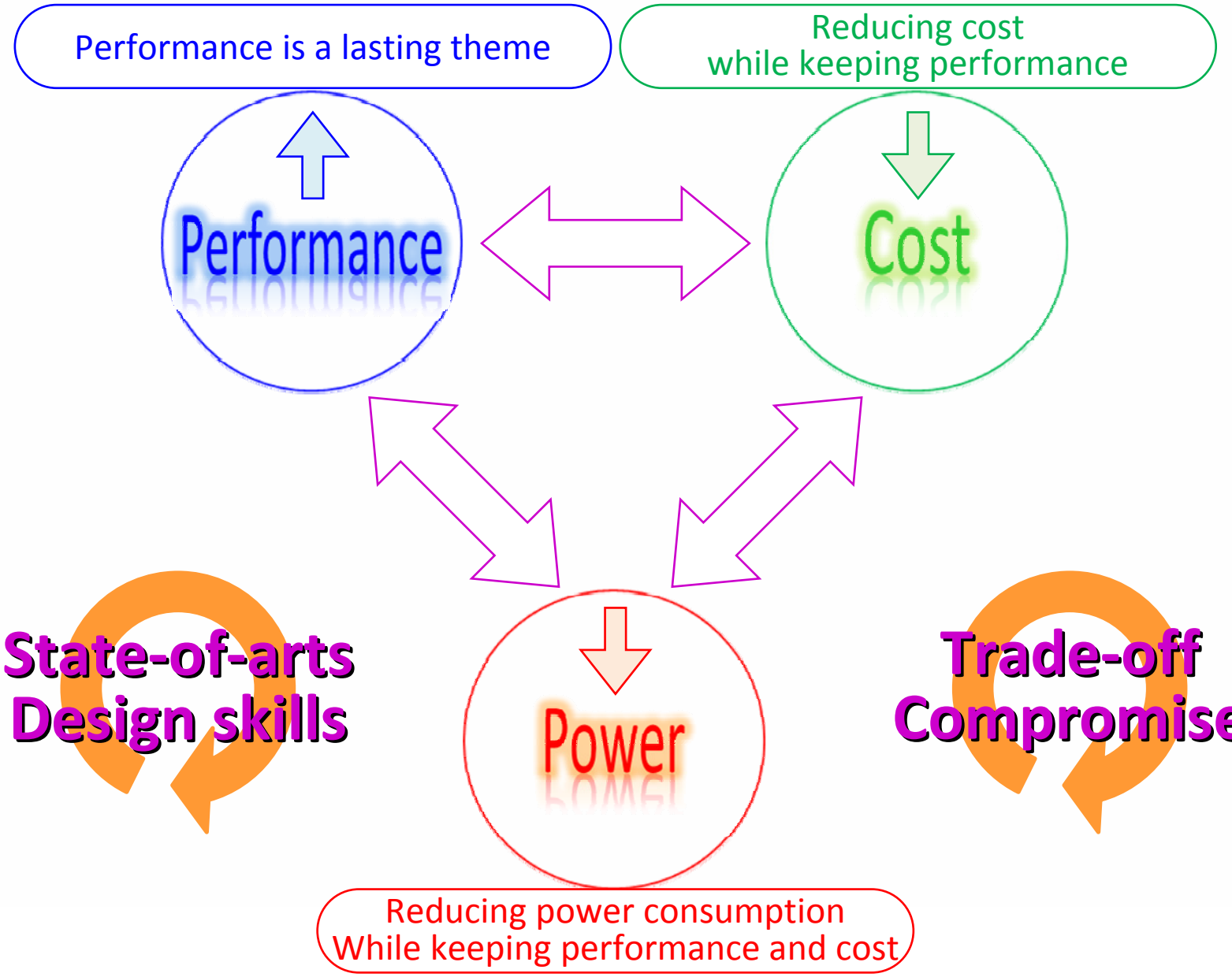
Only a few high-end chip makers today can even afford the exorbitant cost of NEXT-GENERATION RESEARCH AND DESIGN, much less the fabs to build them.

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

Performance, Cost and Power

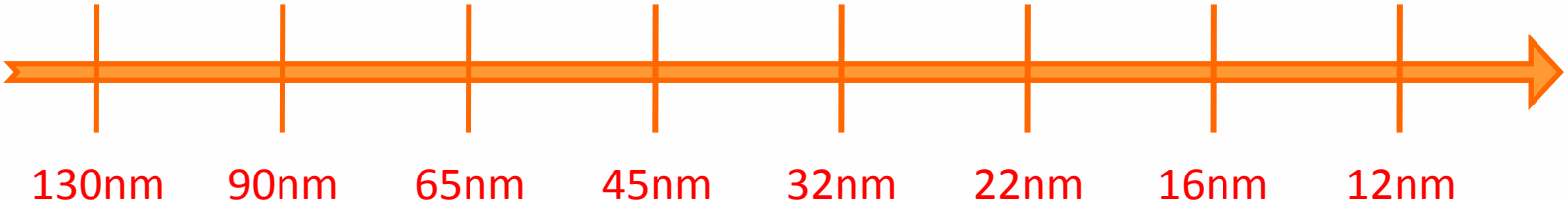


Scaling-down & Von Neumann ISA

High-Performance, Low Power, Low Cost

Scaling-down & Von Neumann Architecture & **Power** Per Chip

Scaling-down & Von Neumann Architecture Per Chip



Summary

- Moore's Low 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

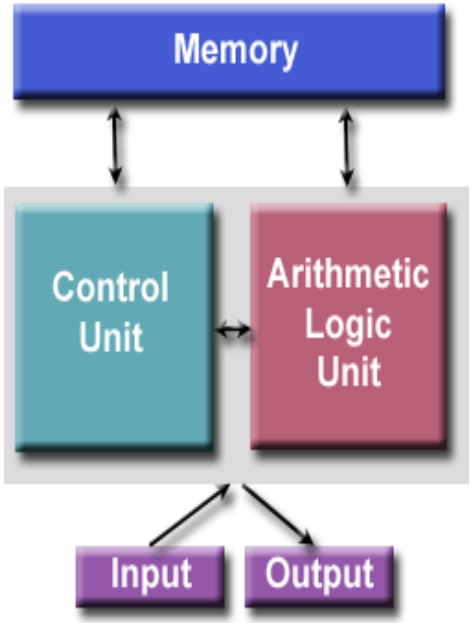
Von Neumann Architecture

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.

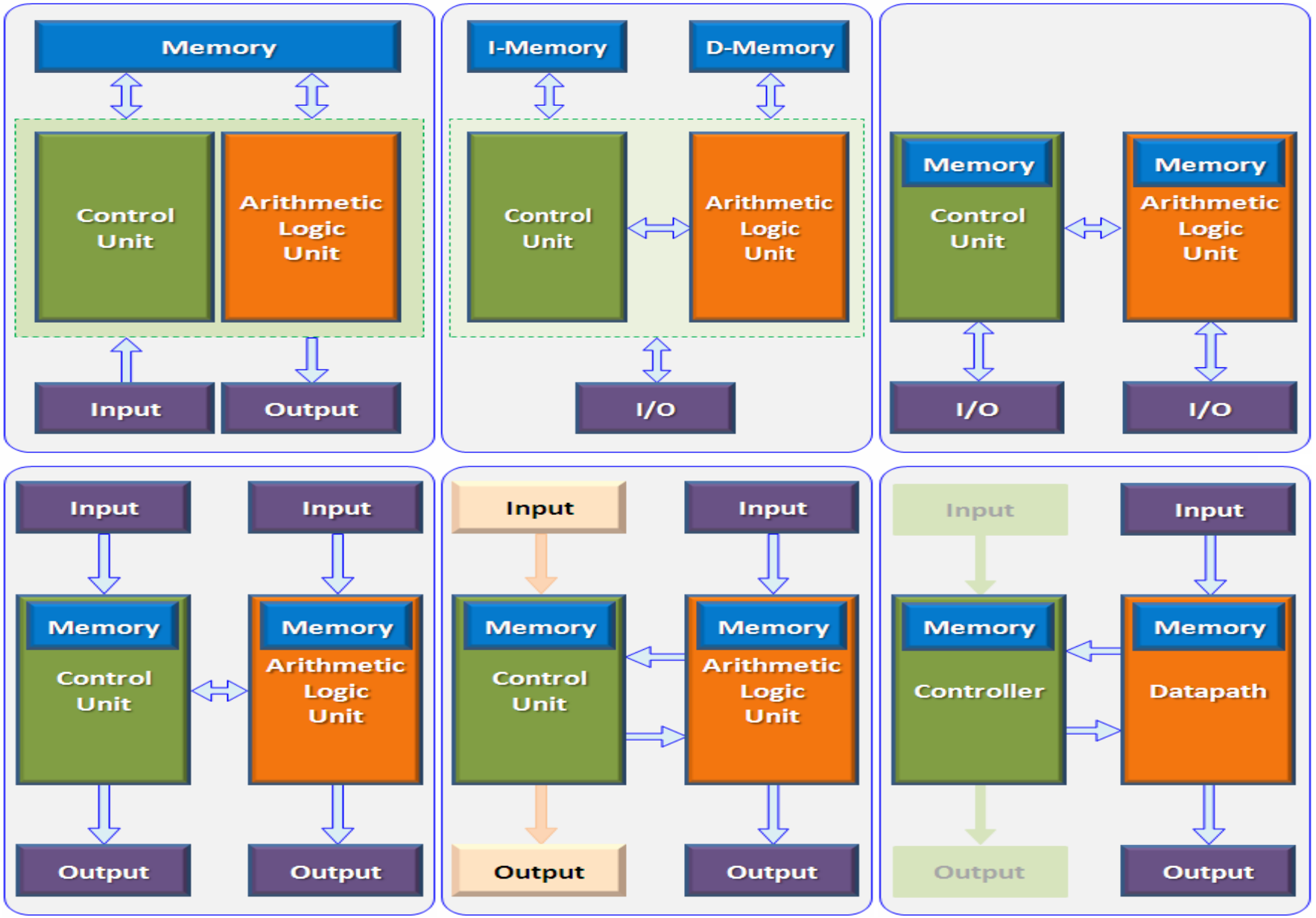
Characteristics of the Von Neumann's architecture:

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**.

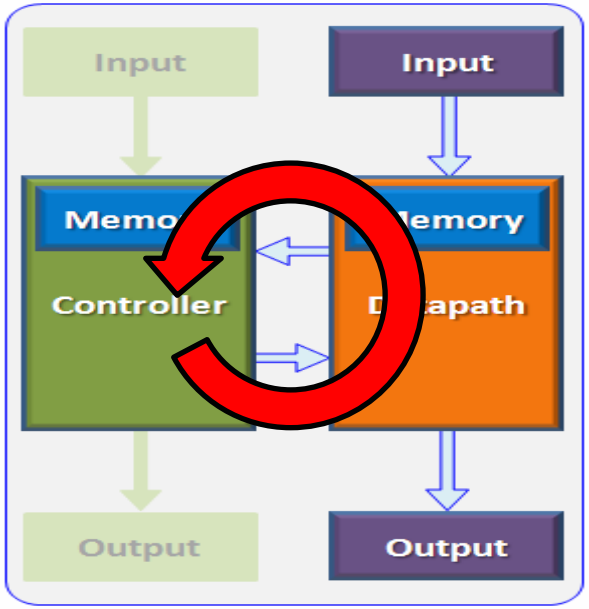


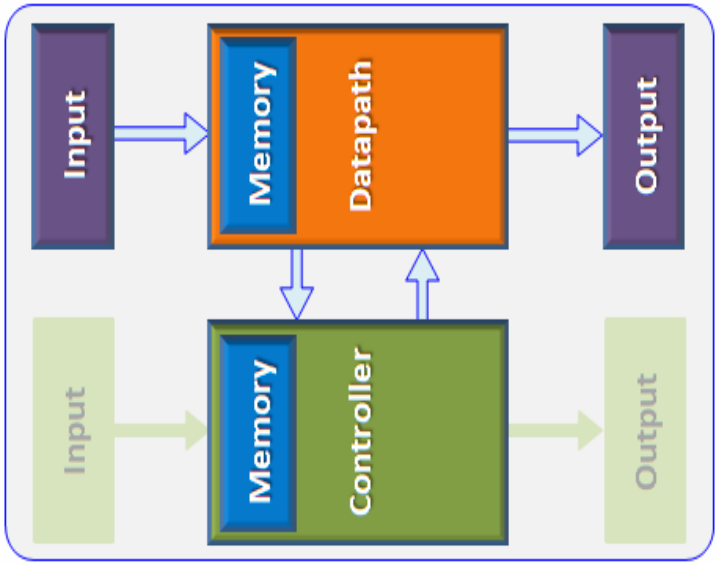
Architecture Evolution

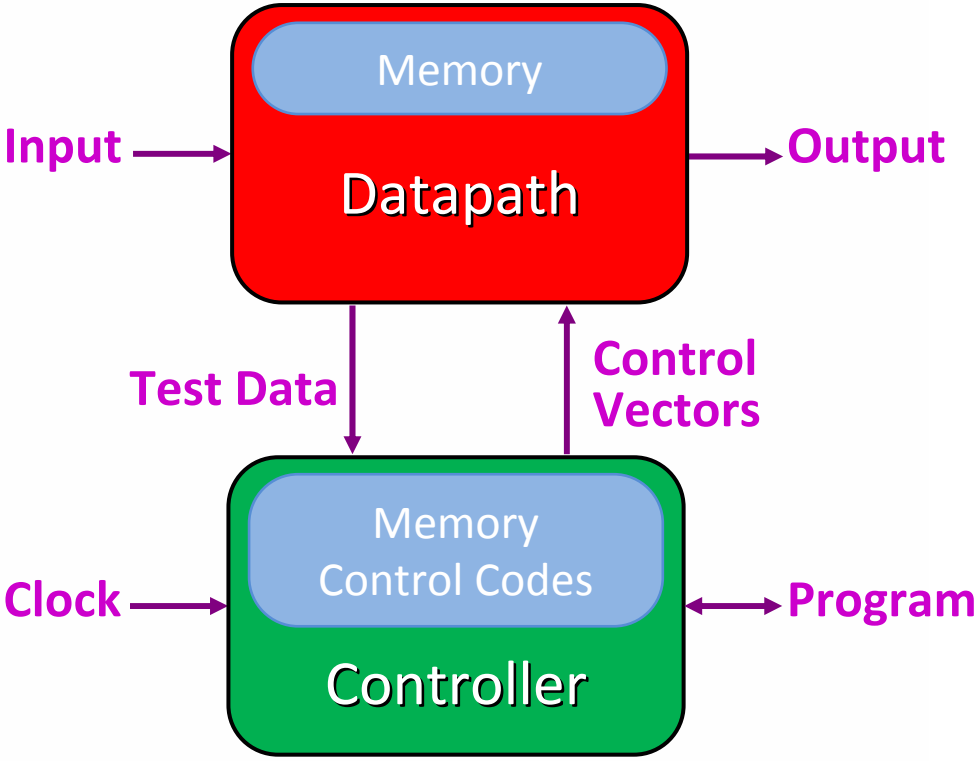


Von Neumann Architecture

Universal Architecture

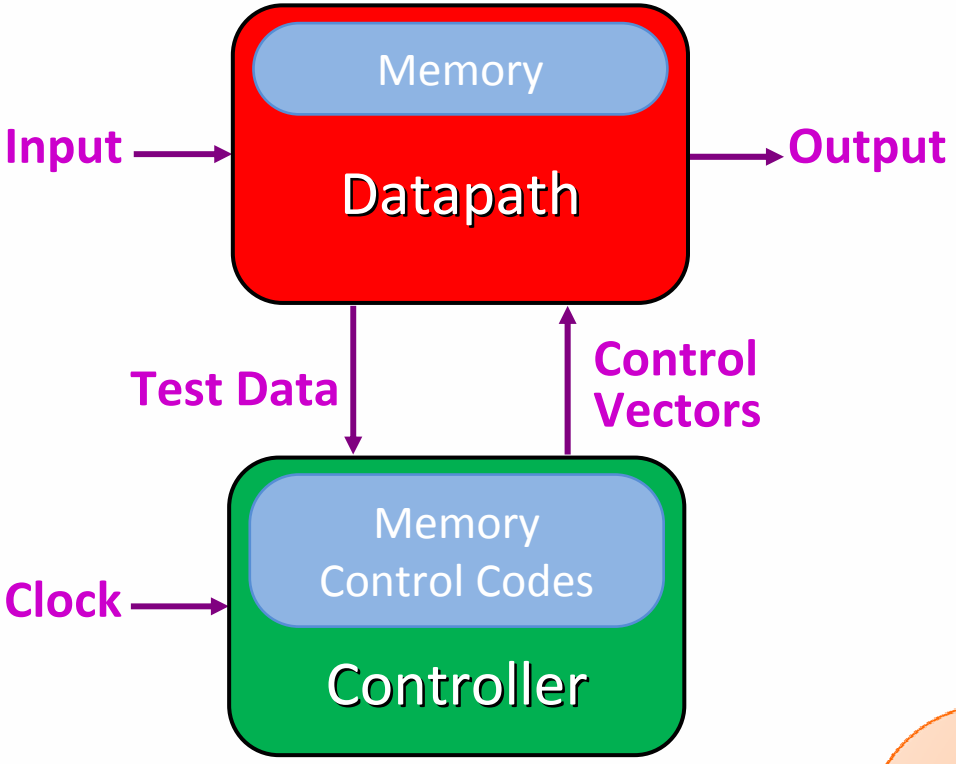






This universal architecture leads to different structures for ASIC and general purpose processor.

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.

- 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.

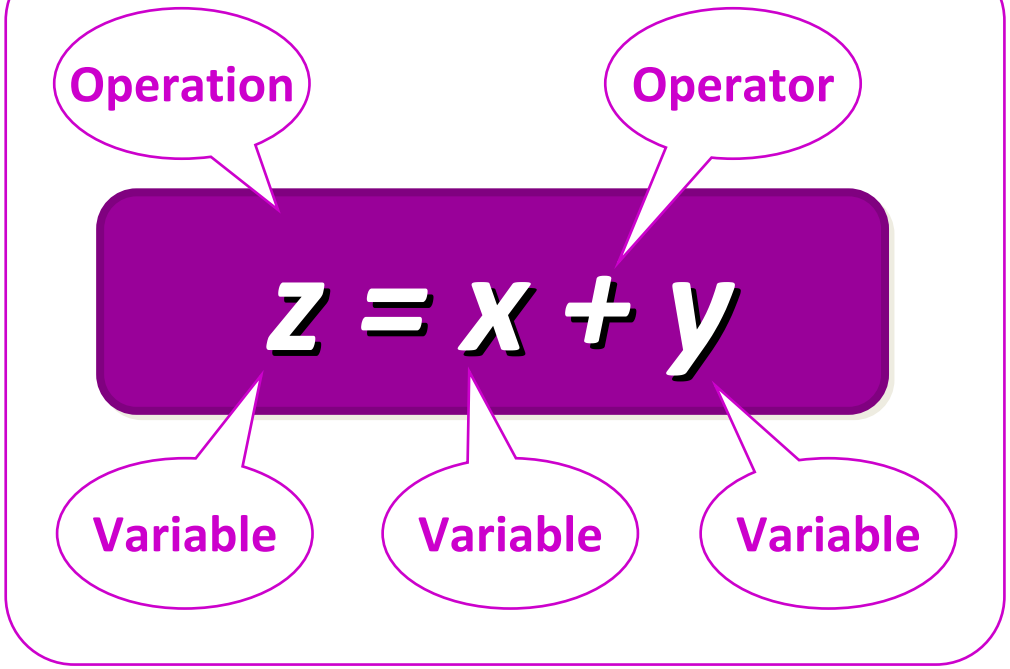
ASP: Application Specific Processor → AISC

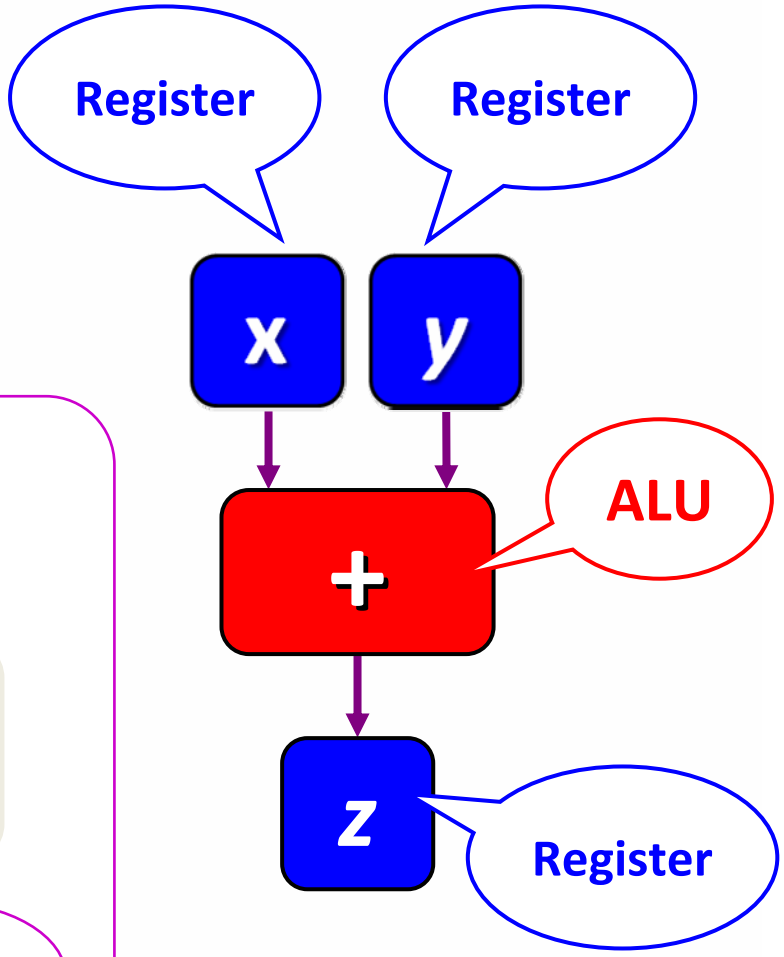
Any digital system can be composed of Datapath and Controller

ASIC: Uniform Architecture

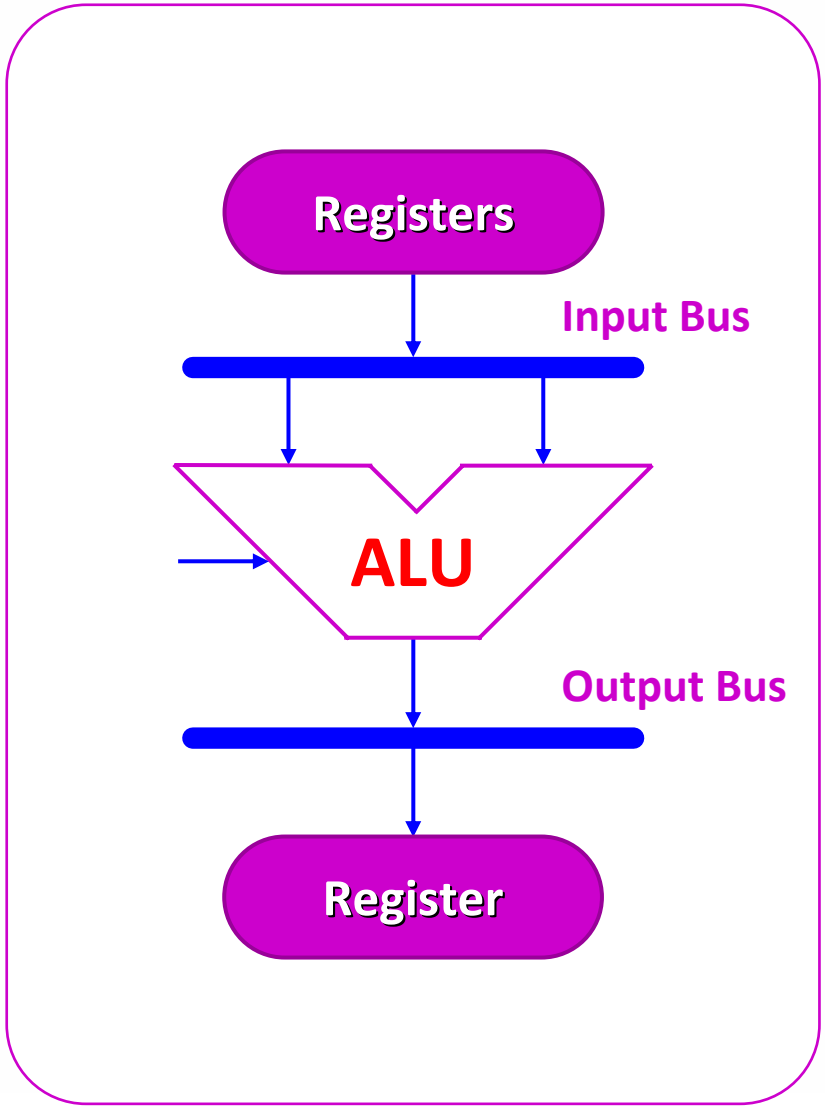

```
SQR(r1) is
BEGIN
   r3 := r1 * 4;
   r3 := r3 + 1;
   r3 := r3 * 0.2;
   FOR i IN INTEGER
      r4 := r1 / r3;
      r3 := r3 + r4;
      r3 := r3 / 2;
   END LOOP;
END;
```

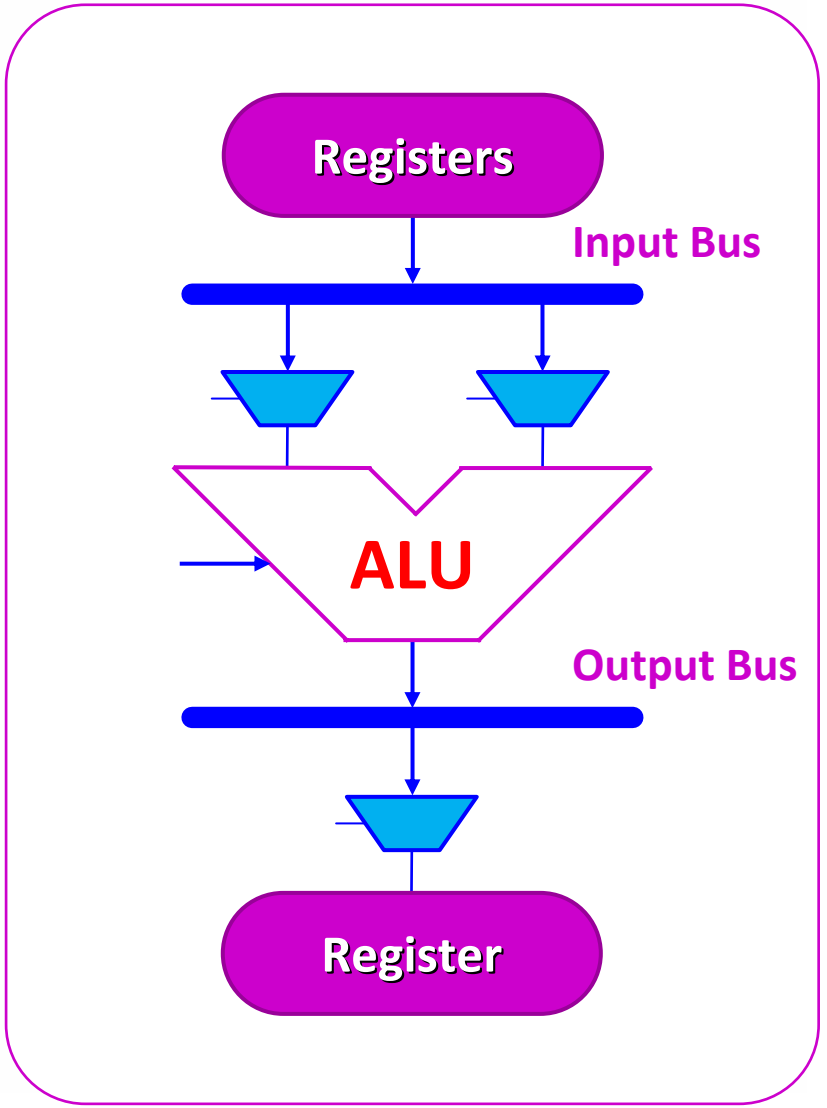
Hardware Descriptions





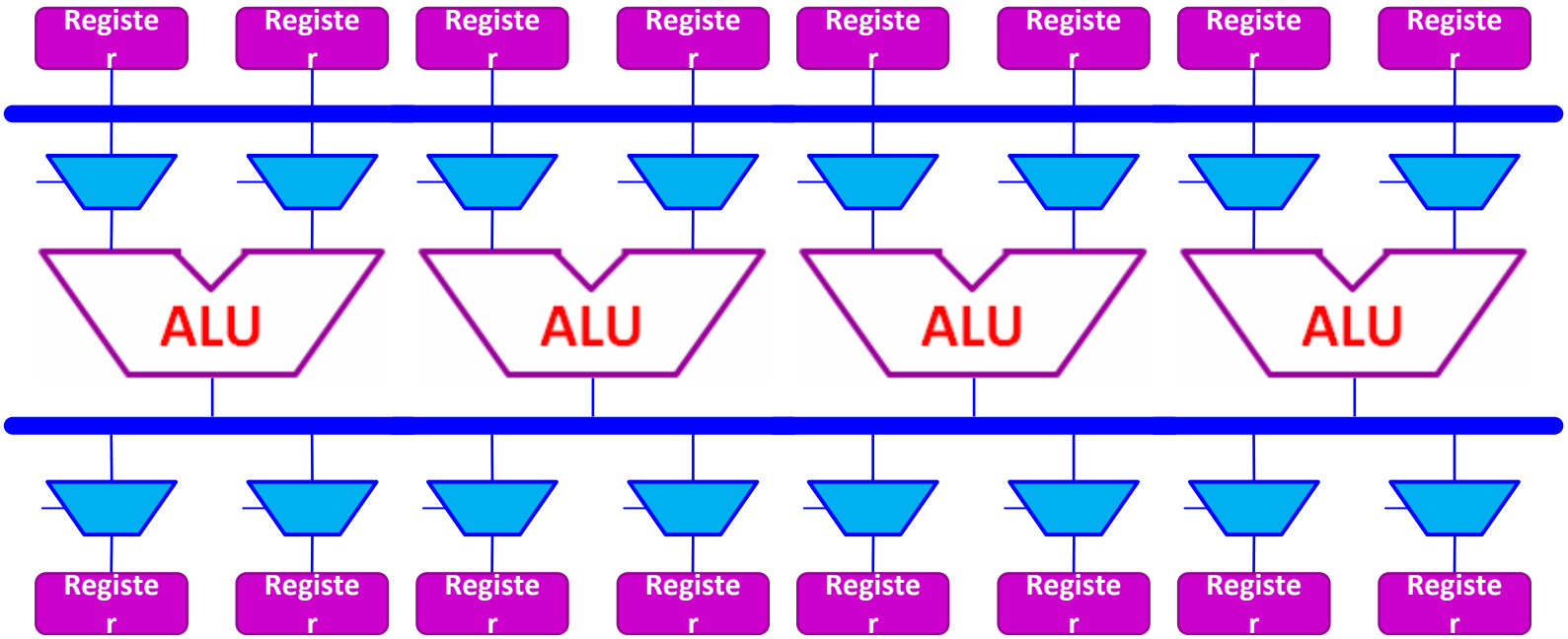
RTL Structure





RTL Level Datapath

1-Dimension Expansion x →



- Parallelizing calculations implied by data dependency.
- Interconnections only exist according to system function.

Hardware: High-Level Synthesis

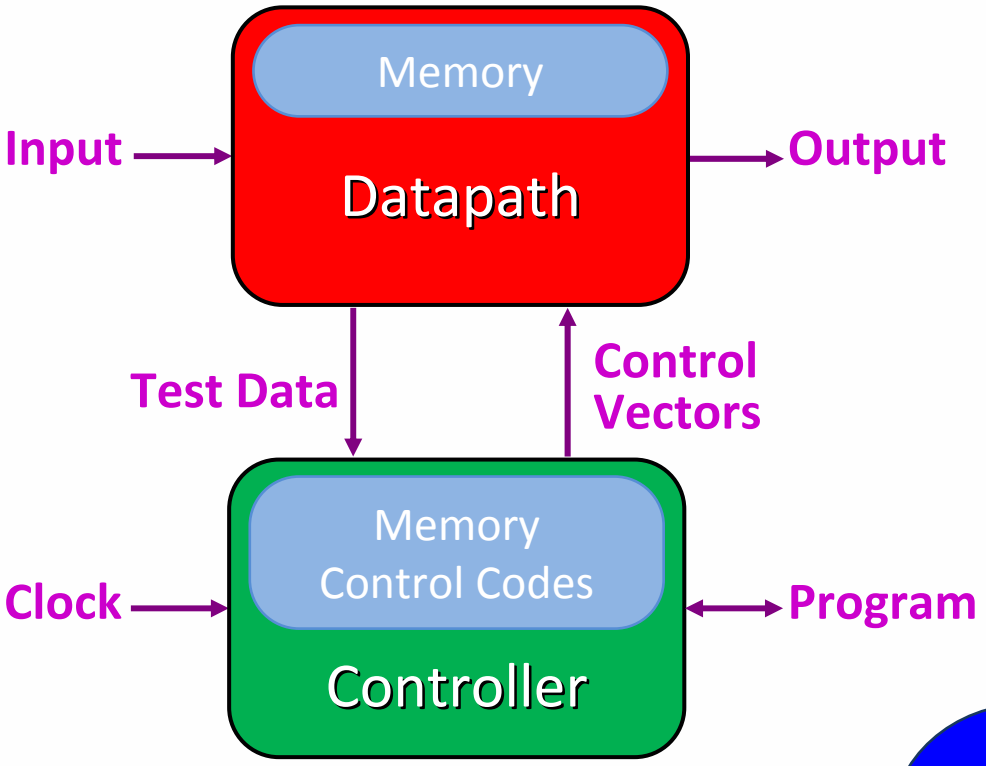

```
SQR(r1) is
BEGIN
   r3 := r1 * 4;
   r3 := r3 + 1;
   r3 := r3 * 0.2;
   FOR i IN INTEGER RANGE 0 TO 3 LOOP
      r4 := r1 / r3;
      r3 := r3 + r4;
      r3 := r3 / 2;
   END LOOP;
END;
```

Hardware Descriptions

- 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

- Operation Scheduling
- Register Optimization
- Resource Allocation
- Interconnection Generation
- Control-Codes Generation
- State Generation
- FSM Design

GPP: 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.

- 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 → MPU

Any processor can also be composed of Datapath and Controller

Software: High-Level Synthesis


```
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 Program

- 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

Operation Scheduling

Register Optimization

Resource Allocation

Interconnection Generation

Control-Codes Generation

State Generation

State Machine Programming

清华大学移动计算研究中心

Summary

- 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.

Outlines

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

Comparisons

Products	Performance	Cost	Power	Flexibility
ASIC	Very High	Very Low	Very Low	Poor
FPGA	Medium	Medium	Medium	Good
CPU	Low	High	High	Very Good
RCP	High	Low	Low	Very Good

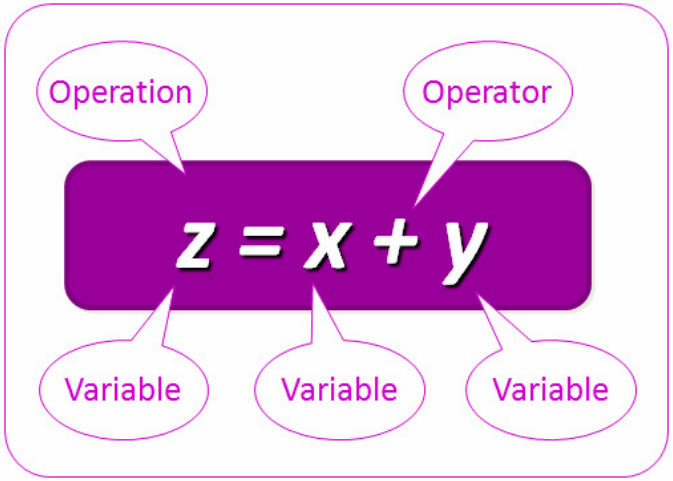
	Performance	MOPS/mm2	nJ/Operation	Programmability
Embedded RISC	Average	<100	~1	Very Good
DSP	Average	<100	~1	Very Good
Multi-Core RISC	High	<100	~1	Good
Multi-Core DSP	High	<100	~1	Good
ASIC	Very High	>1000	~0.01	Poor
RCP	High	~1000	~0.03	Good

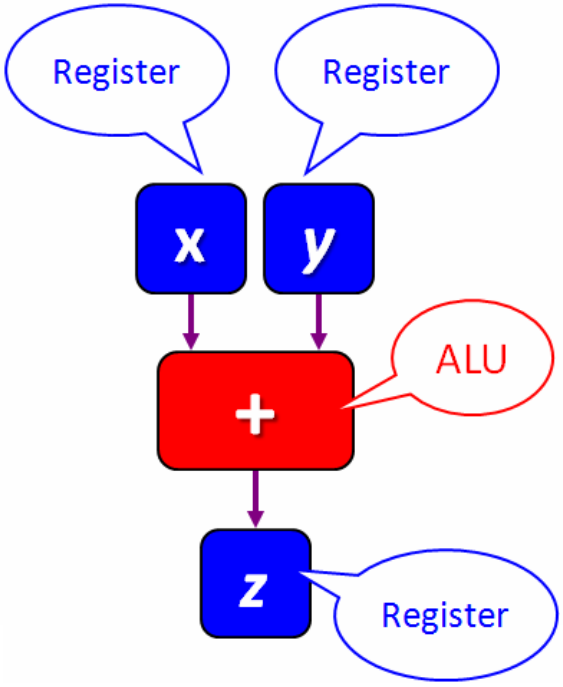
ASIC: Application Specific Integrated Circuit
FPGA: Field Programmable Gate Array
CPU: Central Processing Unit
RCP: Re-Configurable Processor

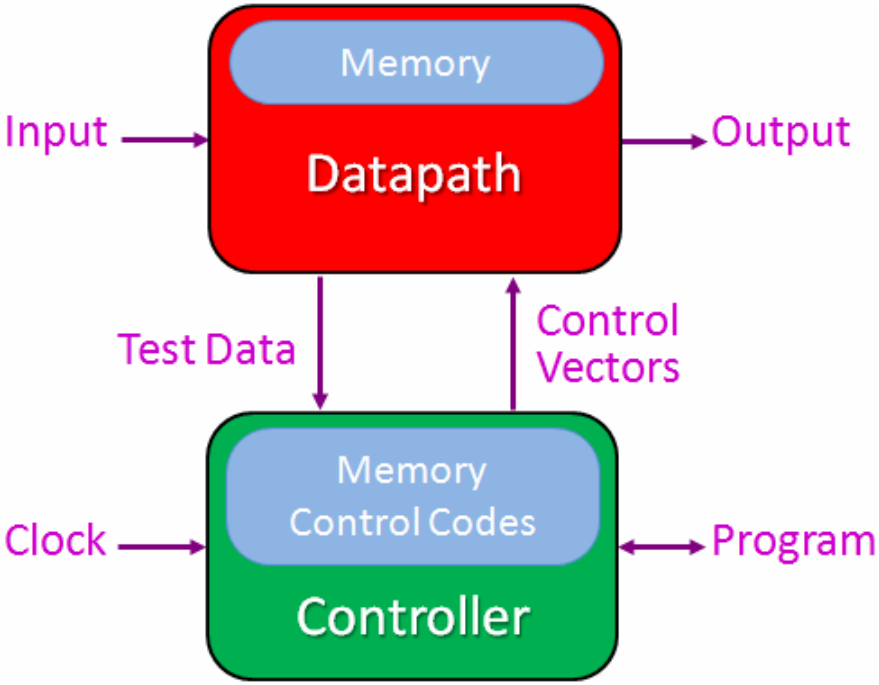
Uniform Architecture

Program

```
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;
```

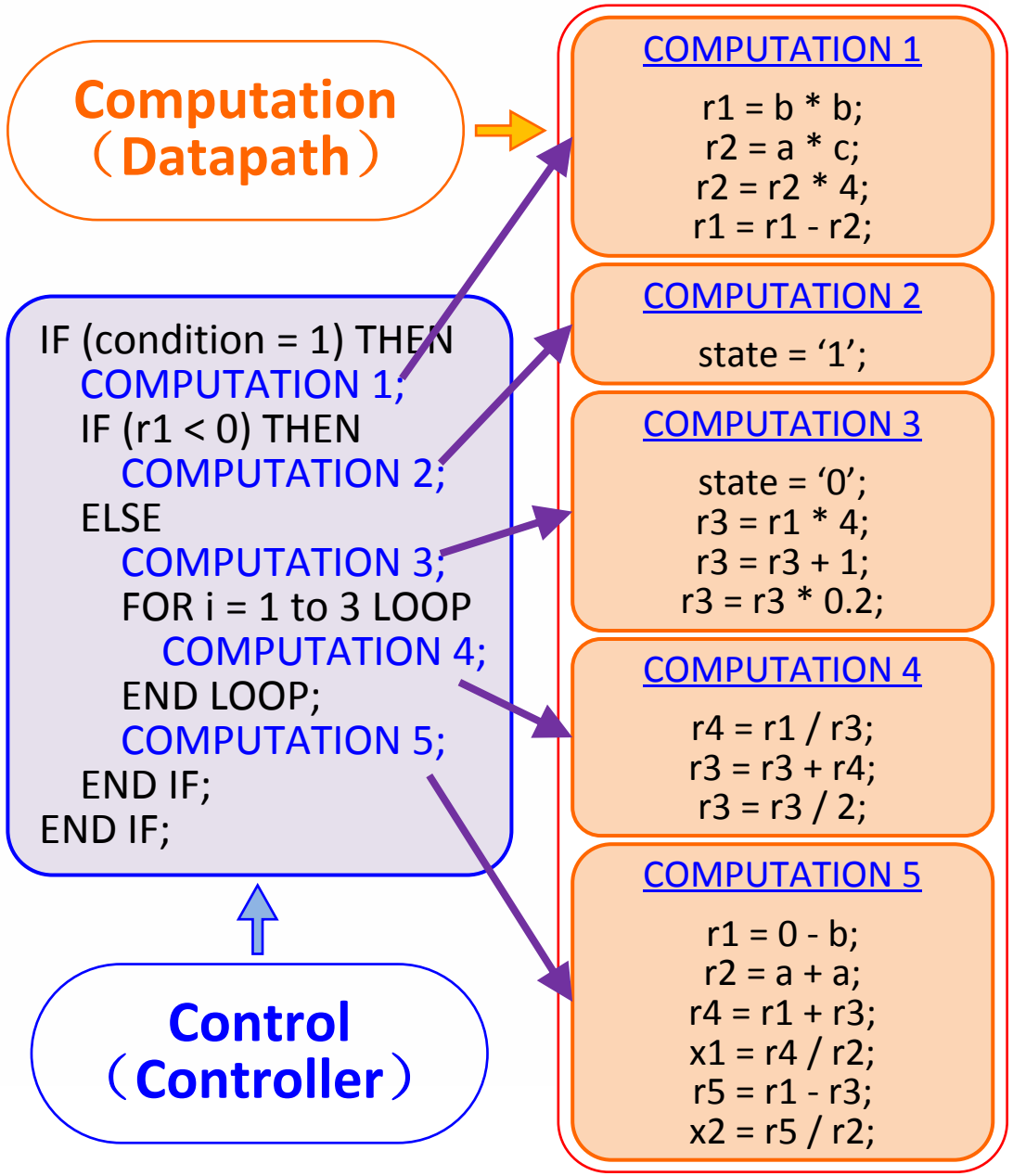






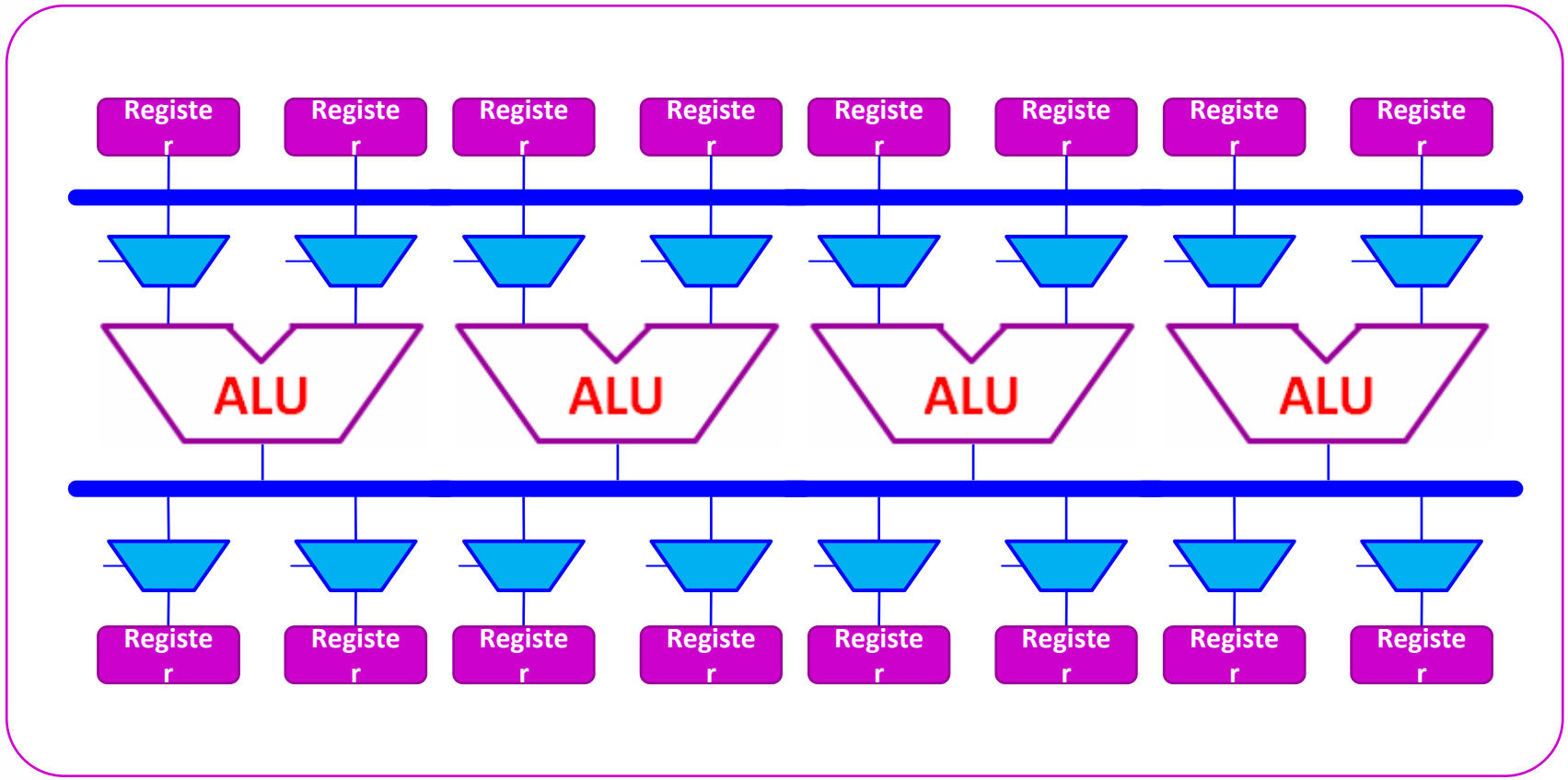
Computation/Control Intensive

```
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;
```



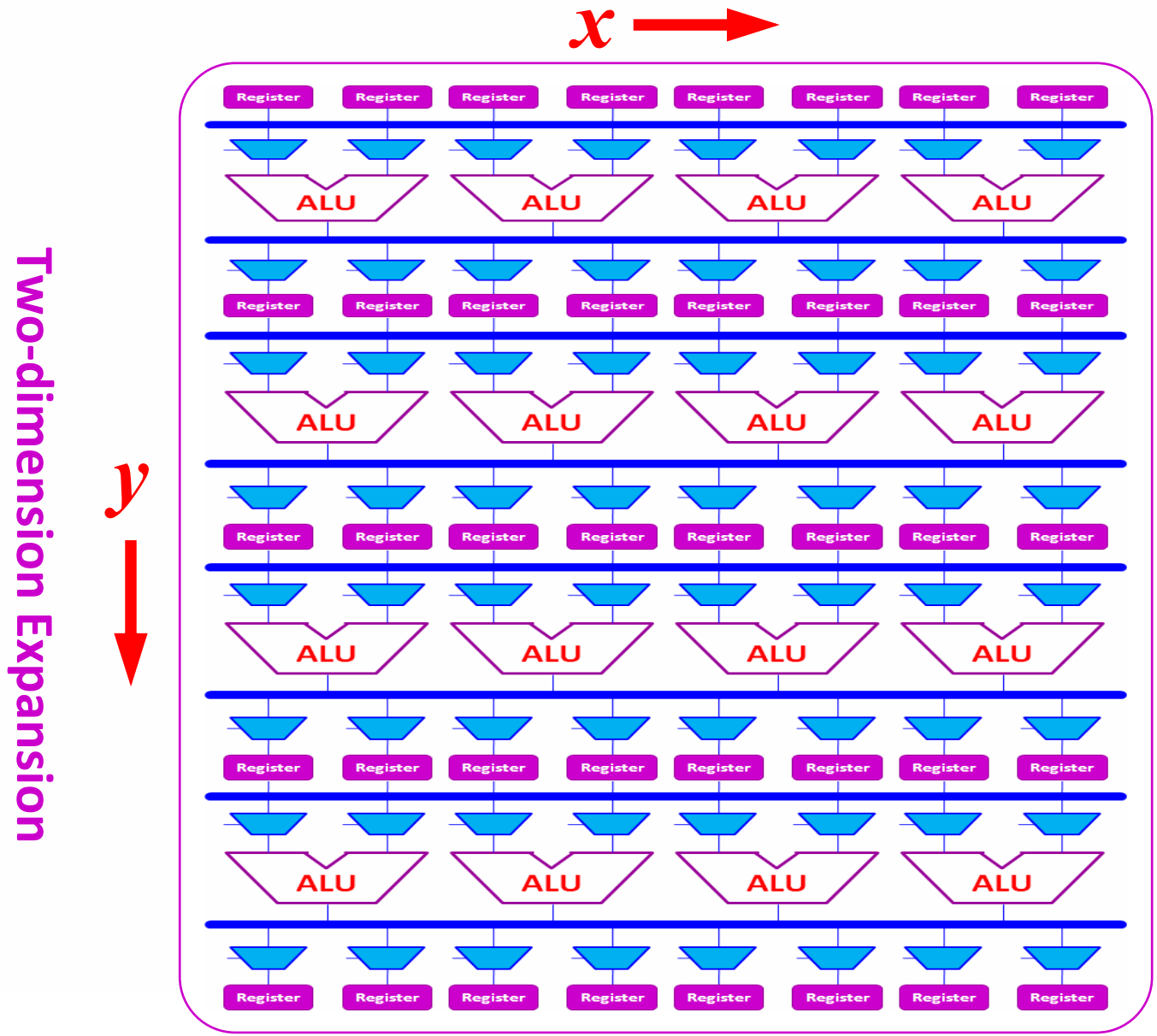
1-Dimension Data-path

1-Dimension Expansion



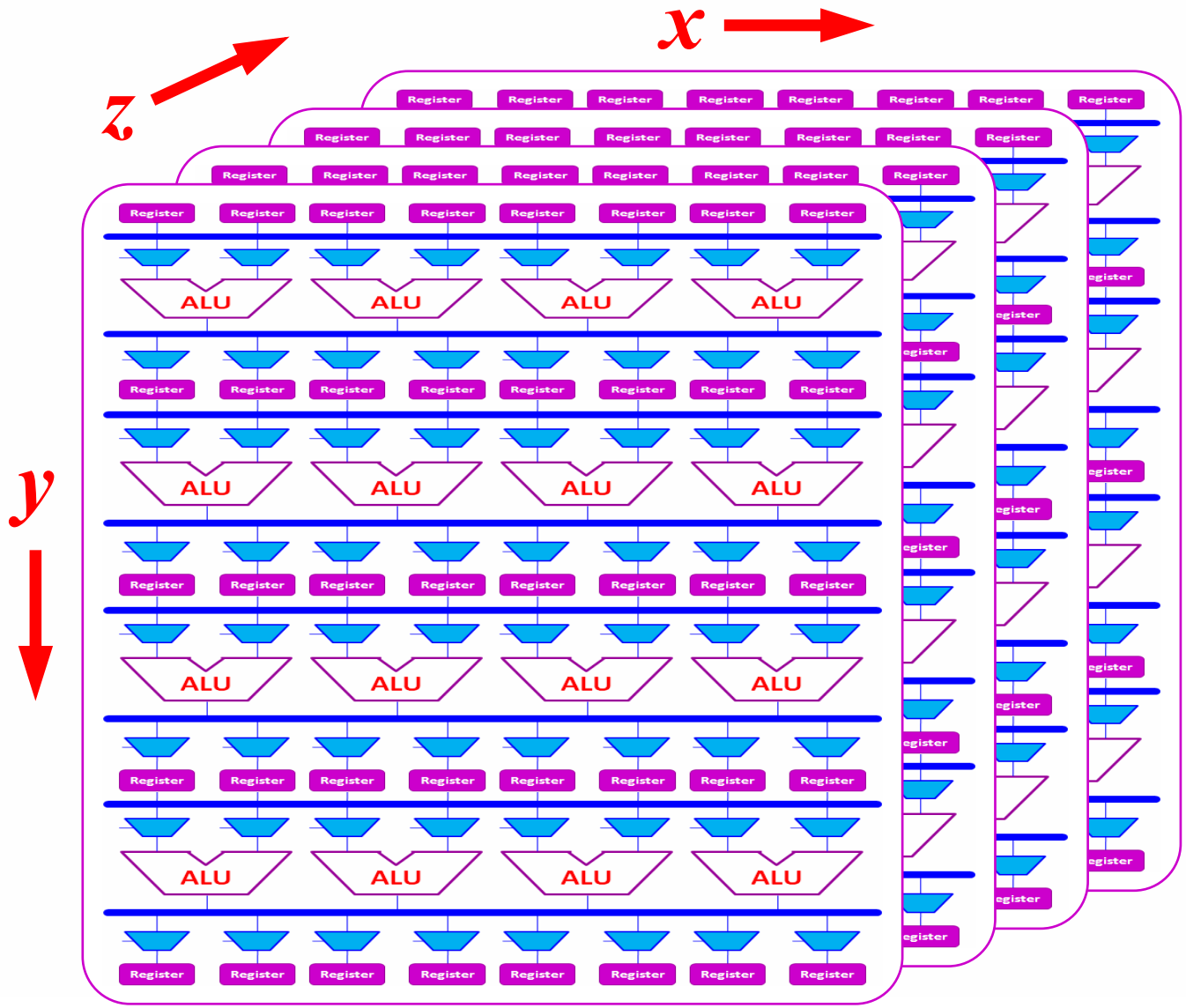
RTL Architecture

2-Dimension Data-path

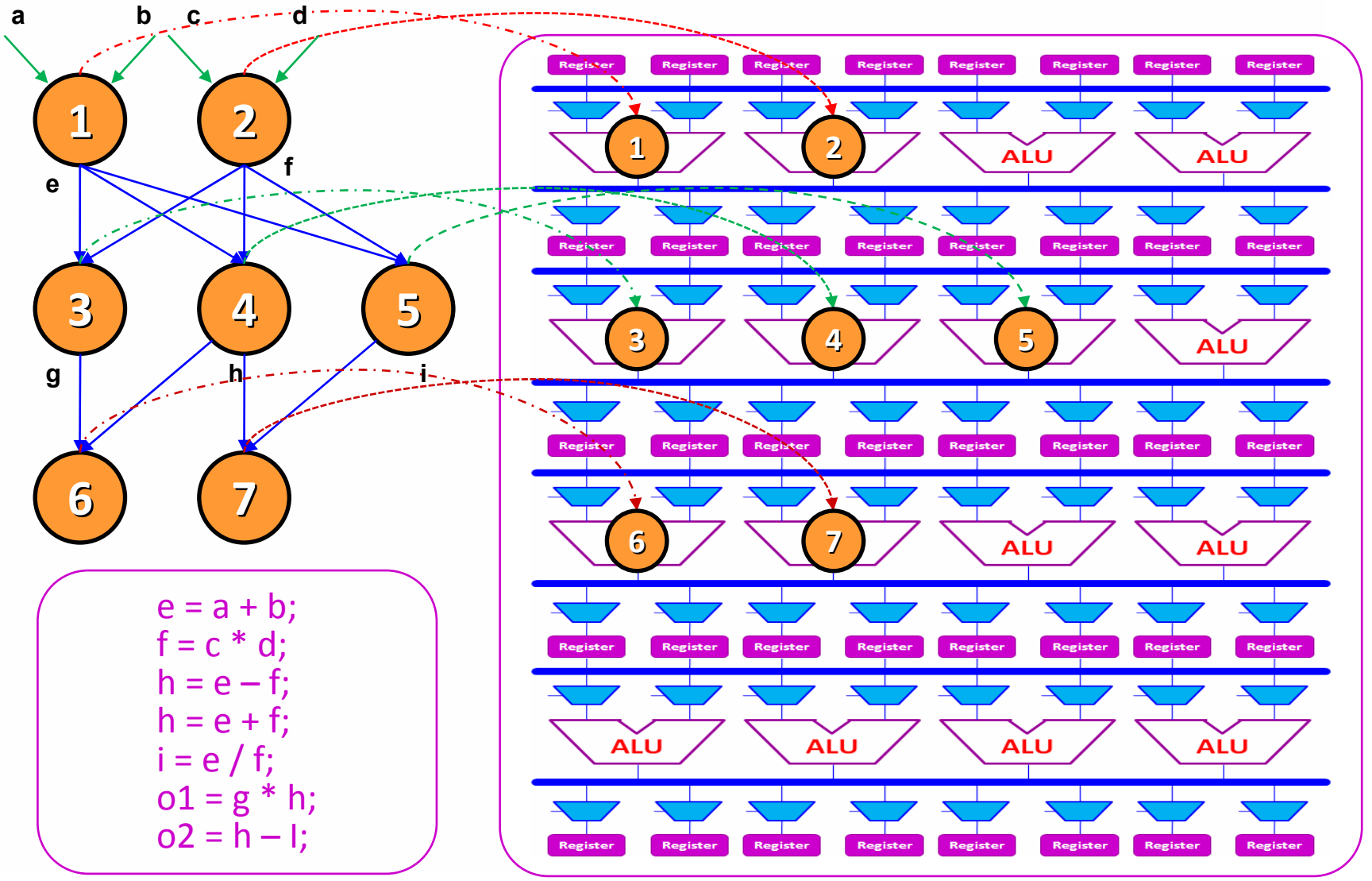


3-Dimension Data-path

Three-dimension Expansion

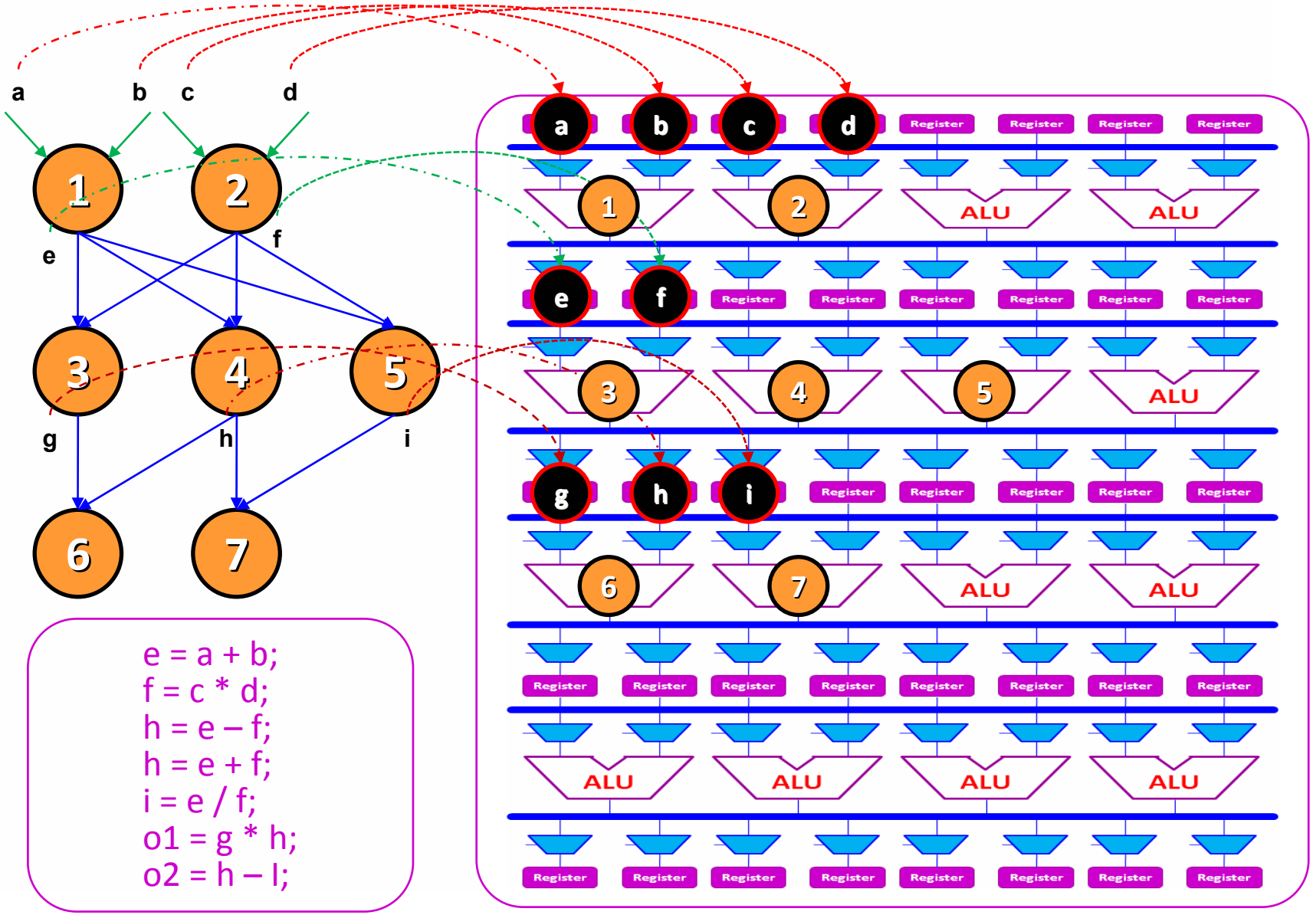


Operations Mapping



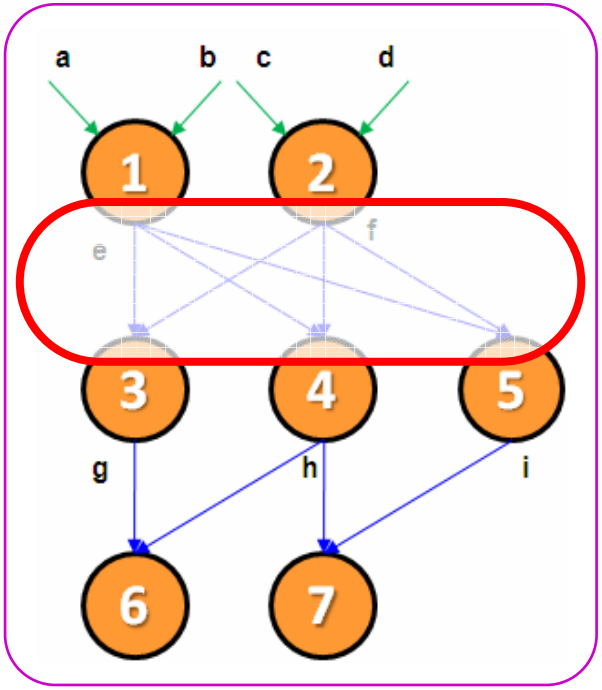

```
e = a + b;
f = c * d;
h = e – f;
h = e + f;
i = e / f;
o1 = g * h;
o2 = h – I;
```

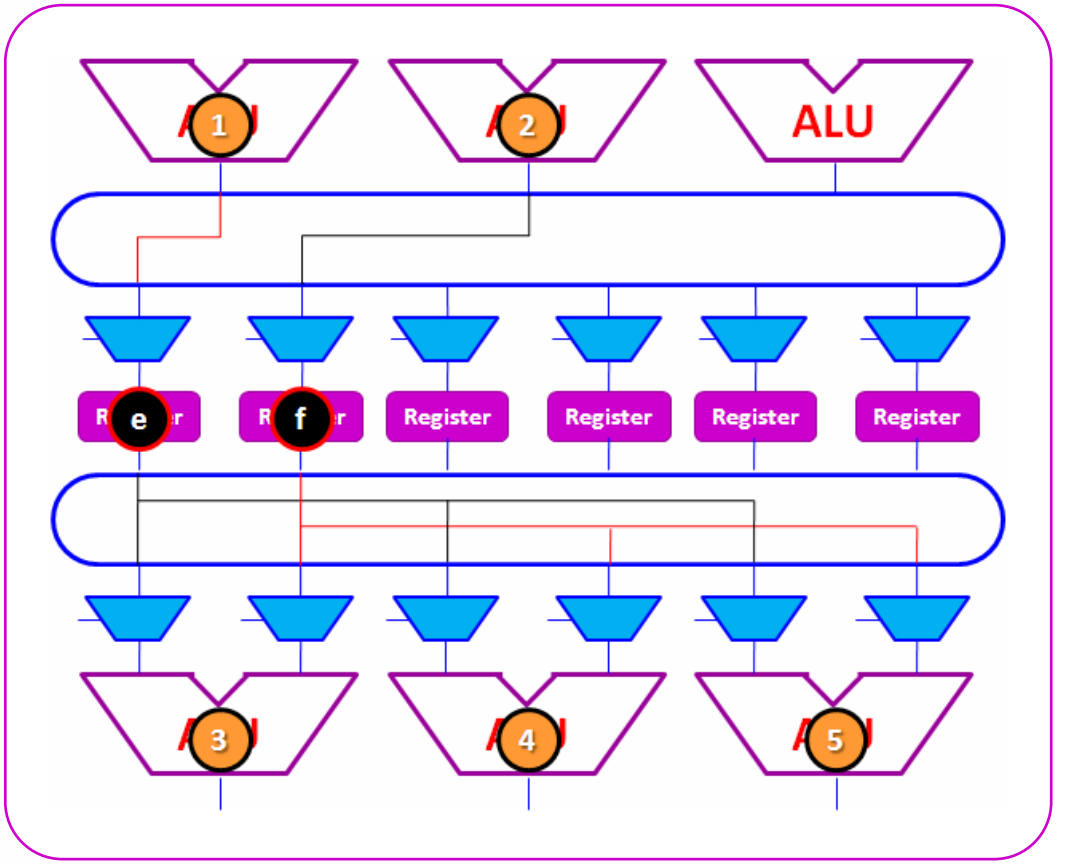
Memories Mapping




```
e = a + b;
f = c * d;
h = e – f;
h = e + f;
i = e / f;
o1 = g * h;
o2 = h – l;
```

Connections Configuration

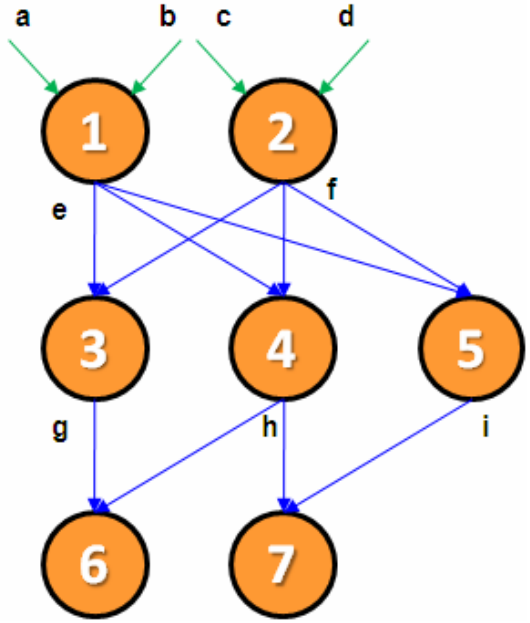


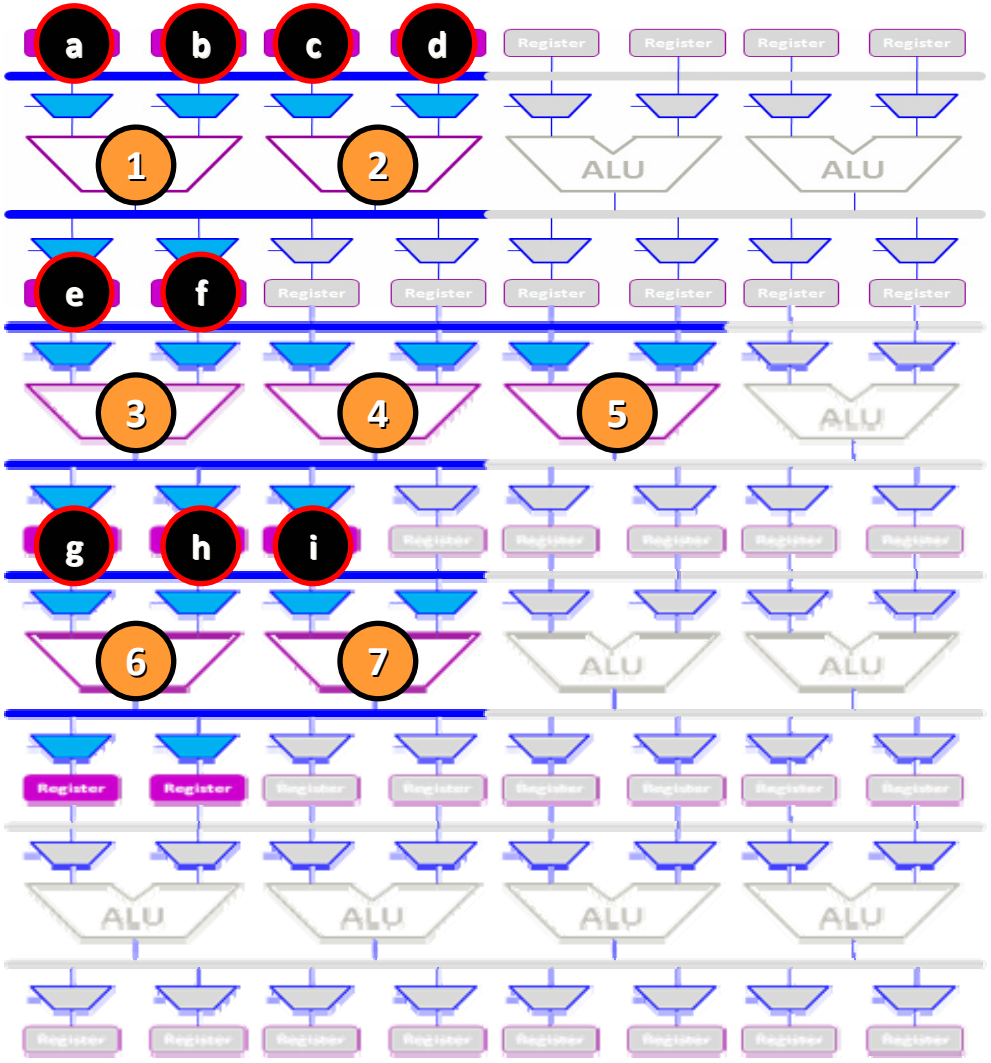


Interconnections among operation 1, operation 2 and operations 3, operation 4, operation 5 are configured according to the data dependency.

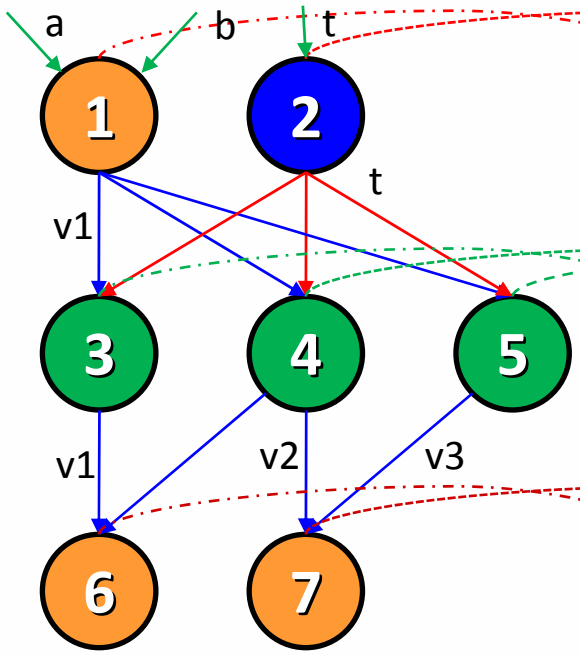
Power-Gating

Processors, memories and interconnections that are not used will be power gated in order to decrease power consumption.

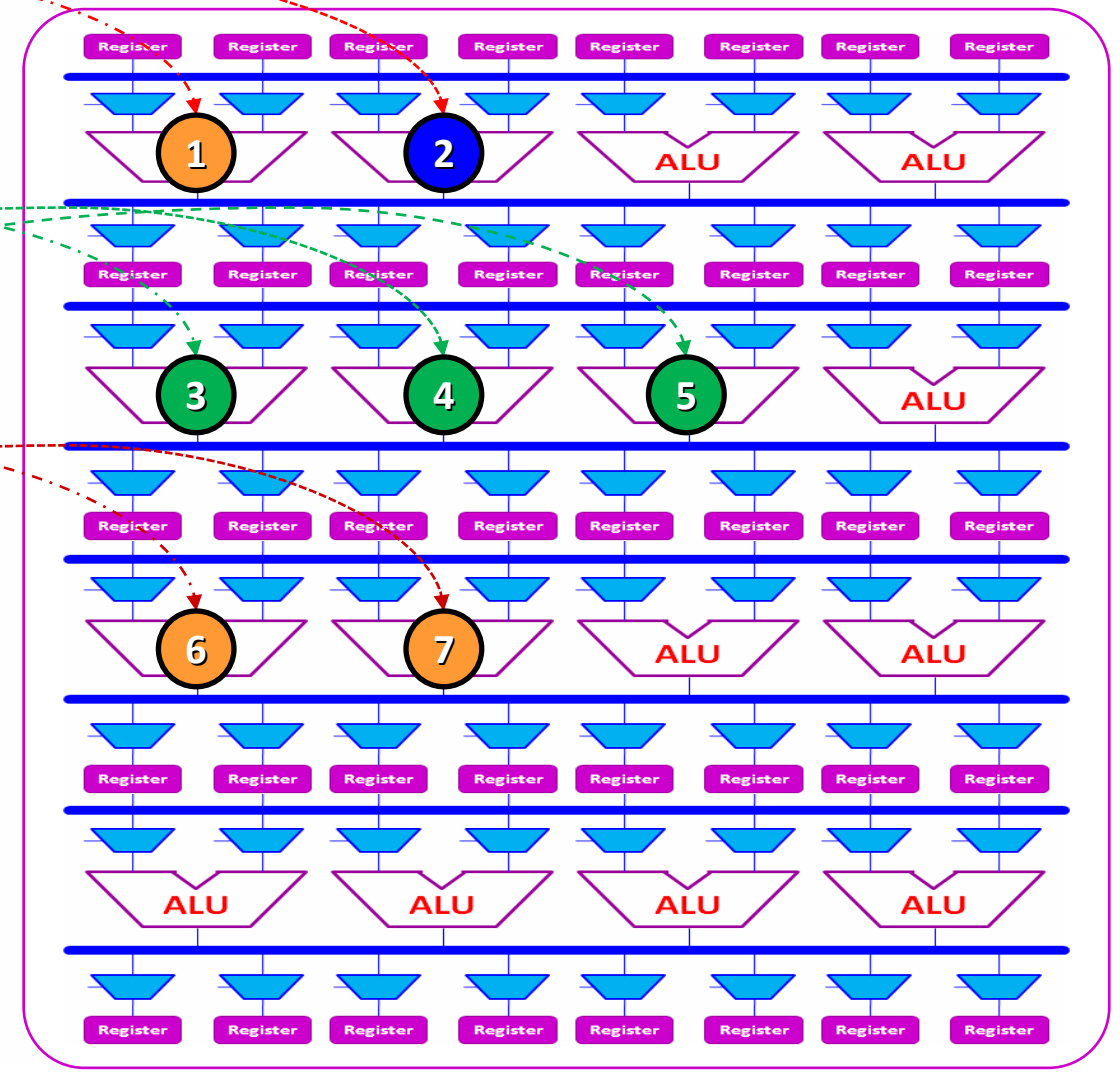




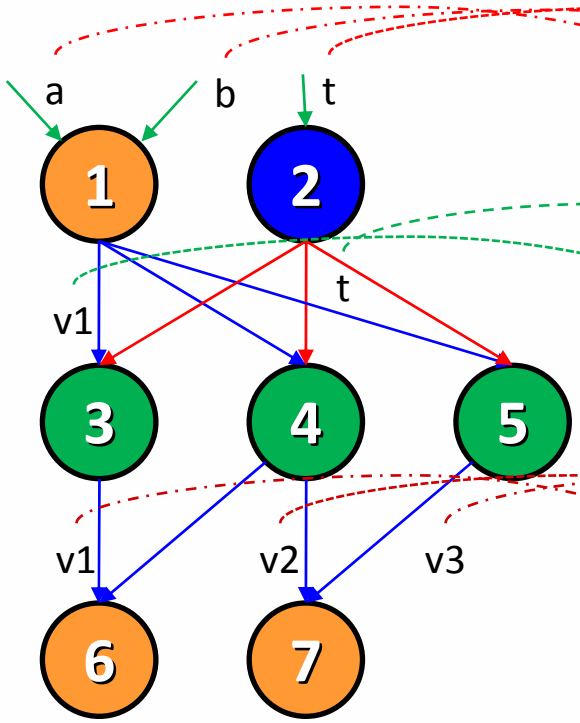
Operations Mapping



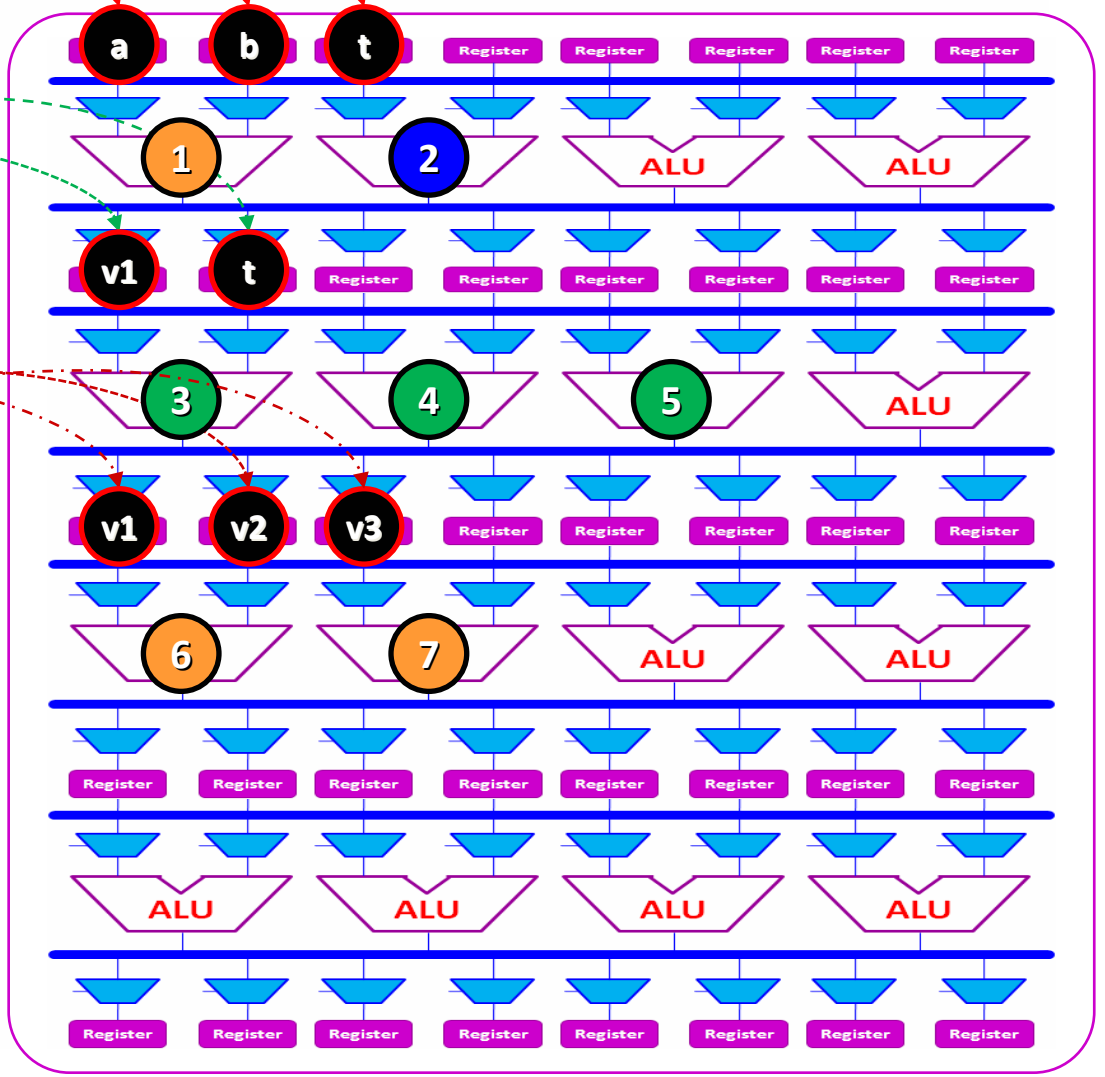

```
v1 = a + b;
Case (t)
1: v1 = v1 * t;
2: v2 = v1 - t;
3: v3 = v1 + t;
End case;
o1 = v1 + v2;
o2 = v2 – v3;
```



Memories Mapping

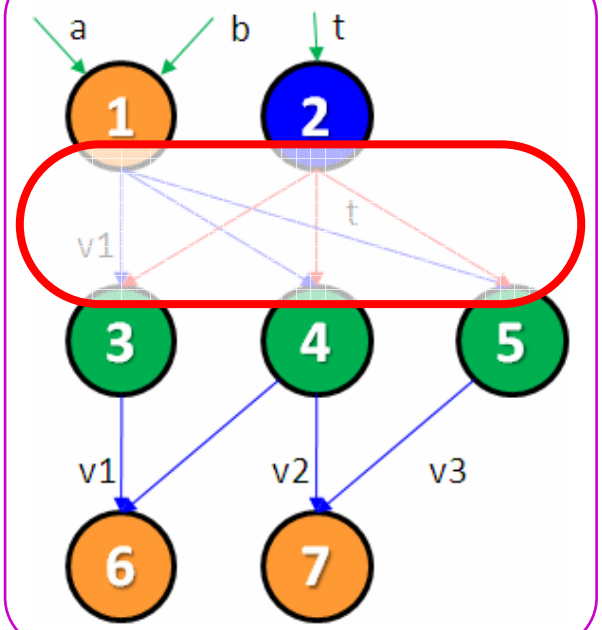


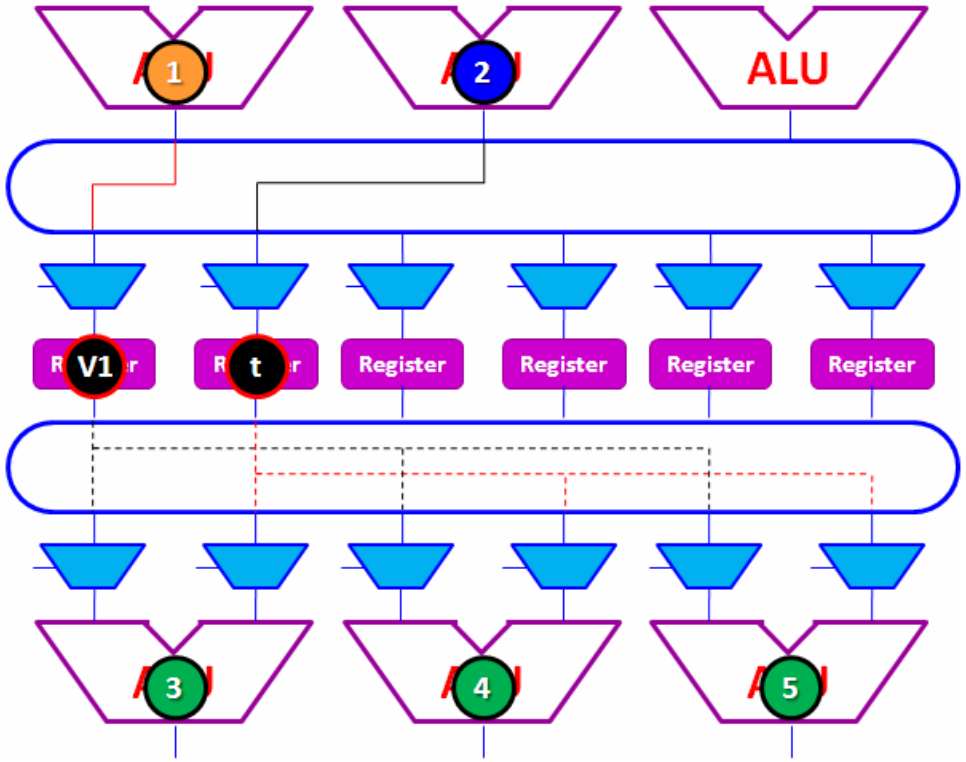

```
v1 = a + b;
Case (t)
1: v1 = v1 * t;
2: v2 = v1 - t;
3: v3 = v1 + t;
End case;
o1 = v1 + v2;
o2 = v2 – v3;
```



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.

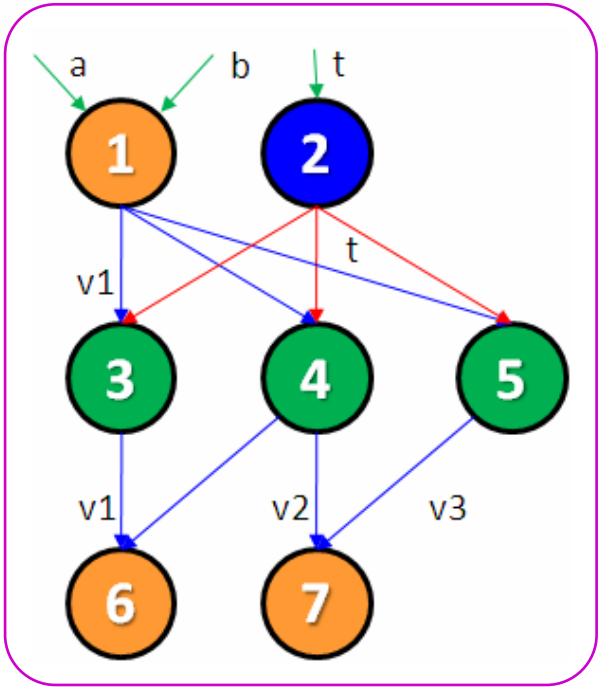


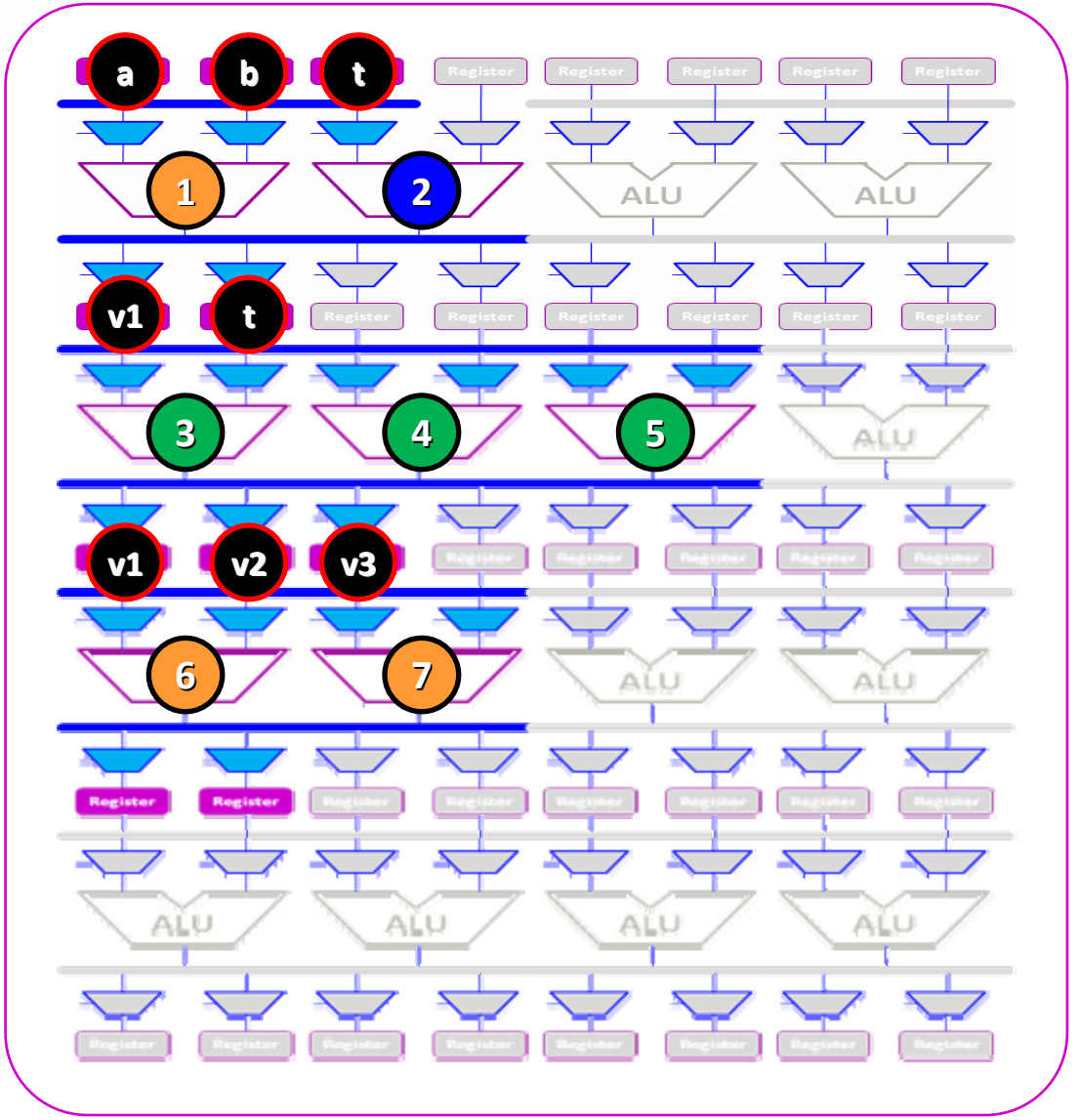


Interconnections among operation 1, operation 2 and operations 3, operation 4, operation 5 are selective depending on the variable t.

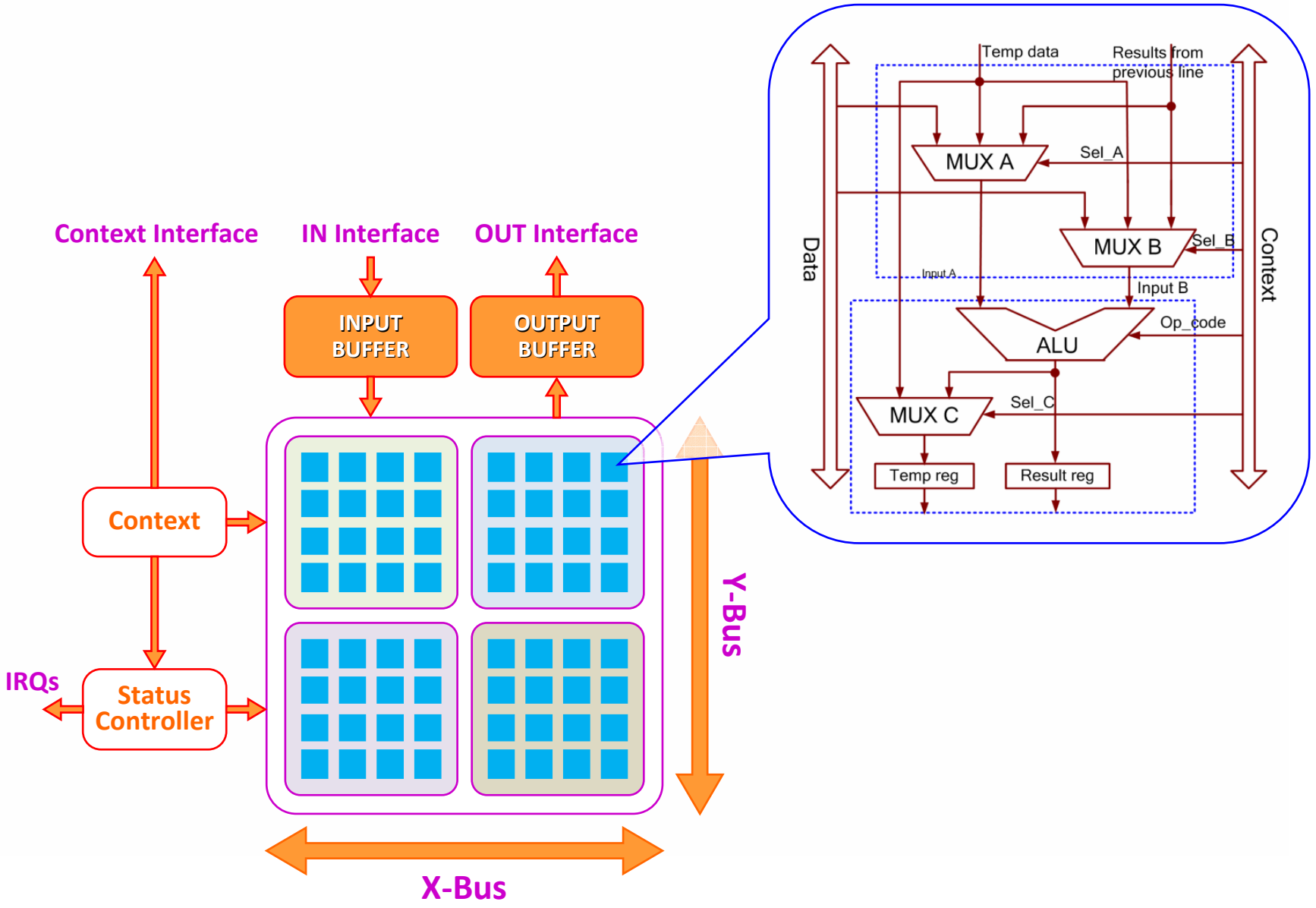
Power-Gating

Processors, memories and interconnections that are not used will be power gated in order to decrease power consumption.





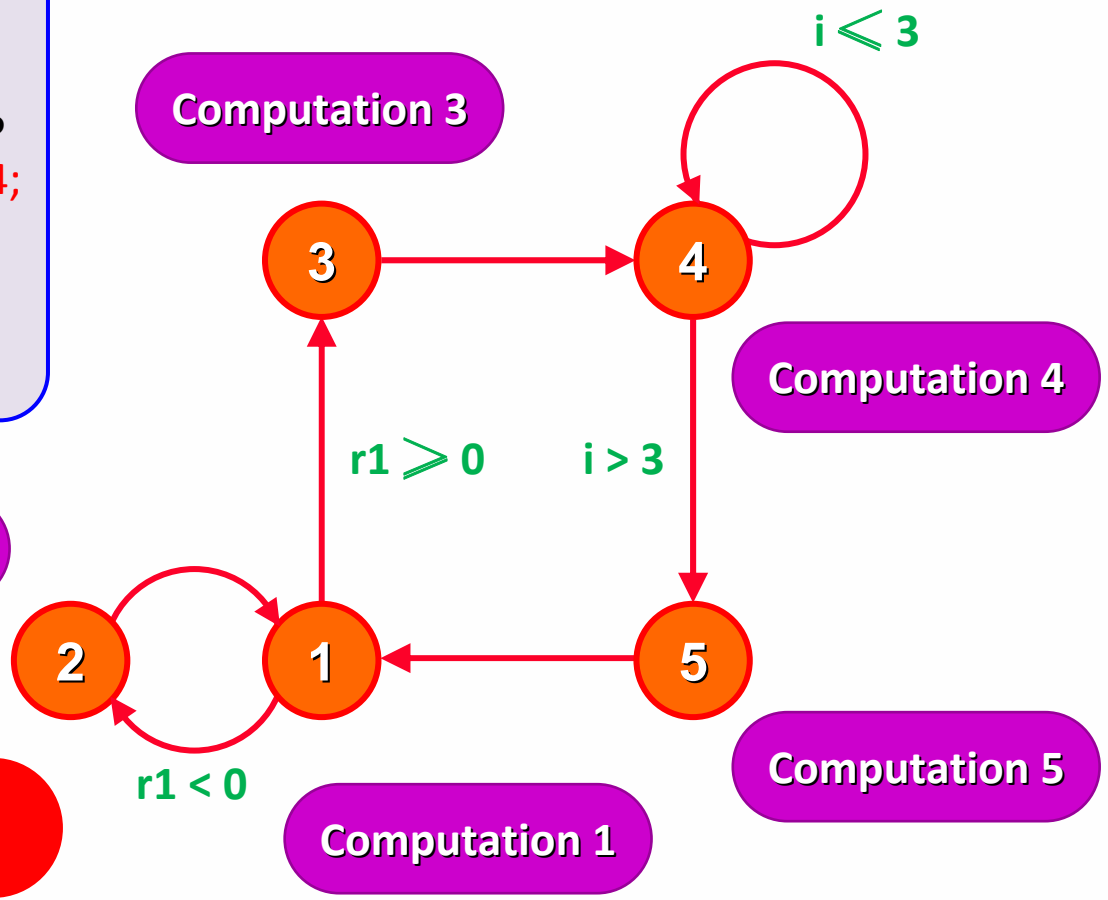
MLU: Multi-Function Unit



Controller: Programmable-FSM


```
IF (condition = 1) THEN
   COMPUTATION 1;
   IF (r1 < 0) THEN
      COMPUTATION 2;
   ELSE
      COMPUTATION 3;
      FOR i = 1 to 3 LOOP
         COMPUTATION 4;
      END LOOP;
      COMPUTATION 5;
   END IF;
END IF;
```

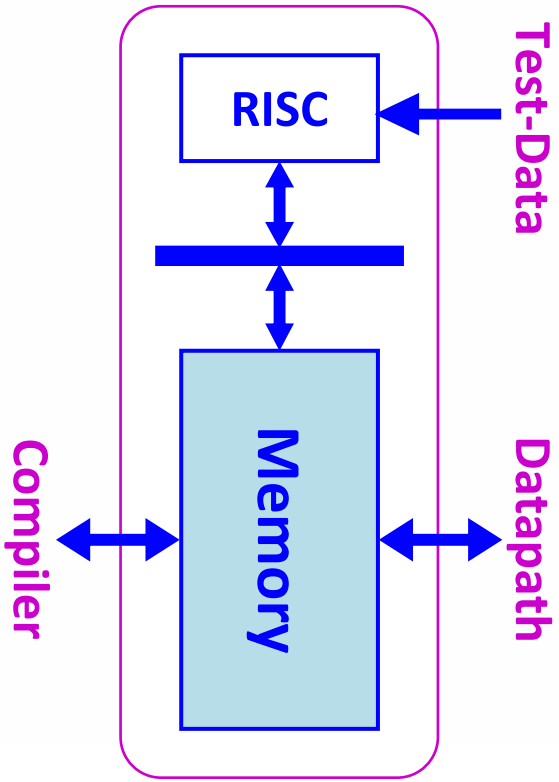
FSM: Finite State Machine

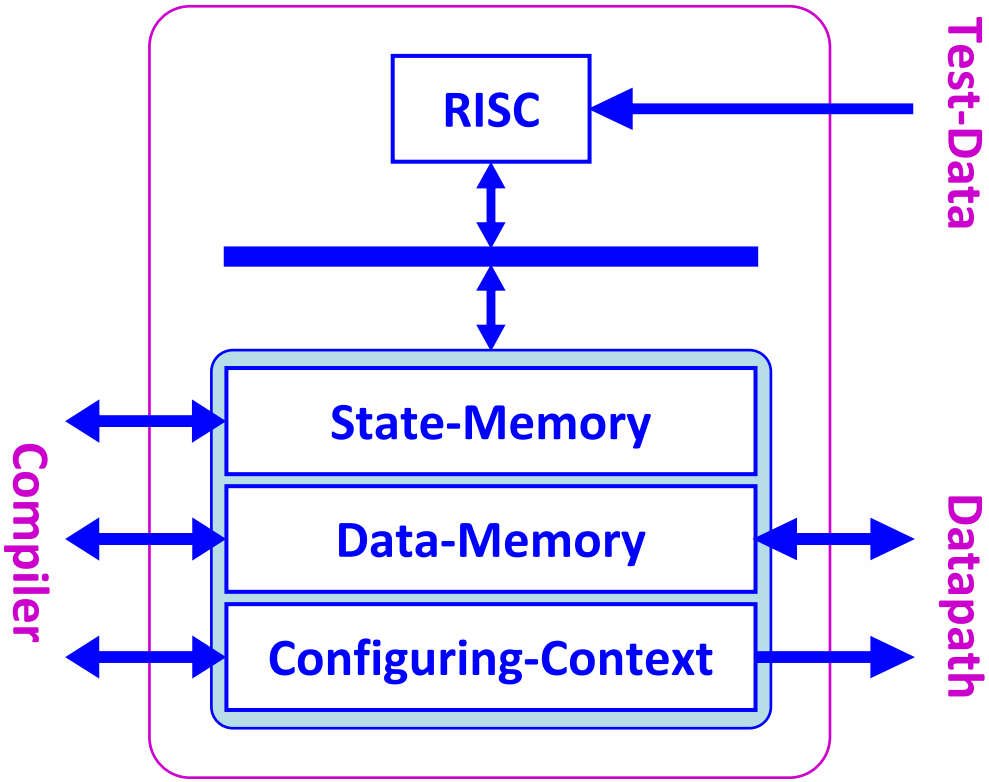


Programmable

RISC Based Programmable-FSM

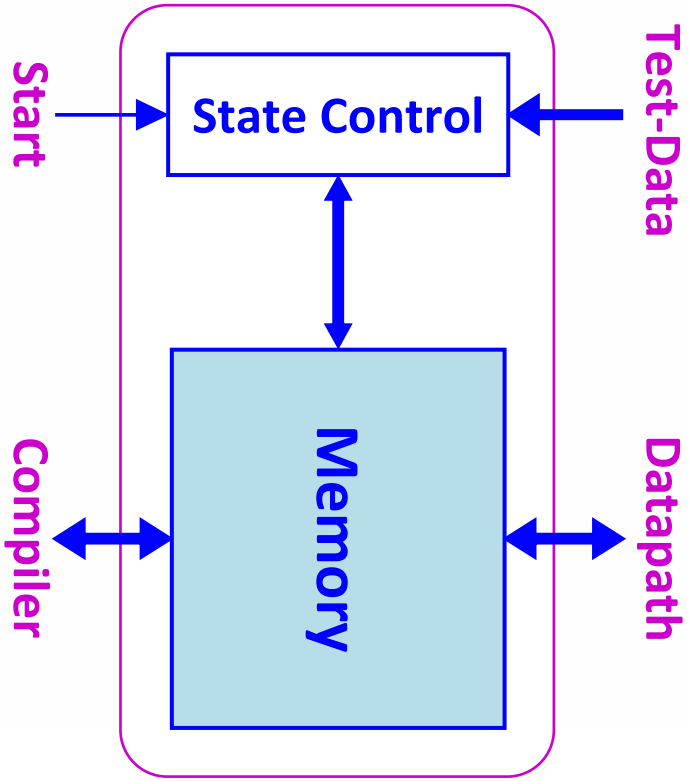
RB-PFSM:
1. Read current state from State-Memory
2. Decide next state and generate address according to test data, that address the Data-Memory and Configuring-Context
3. Output the data and configuring-context

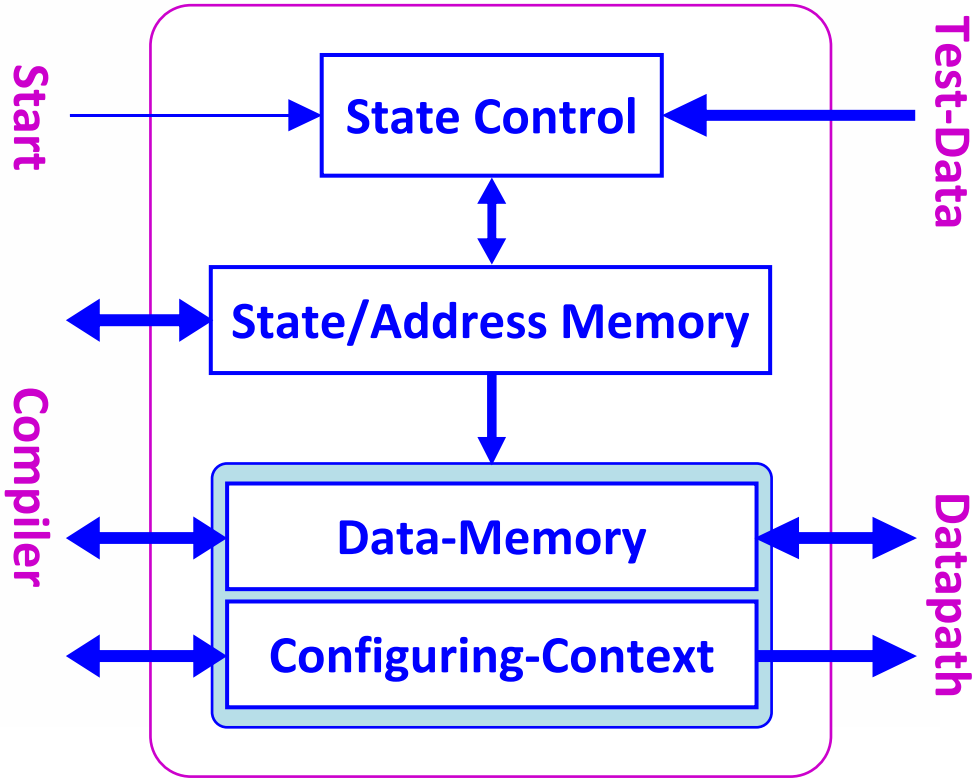




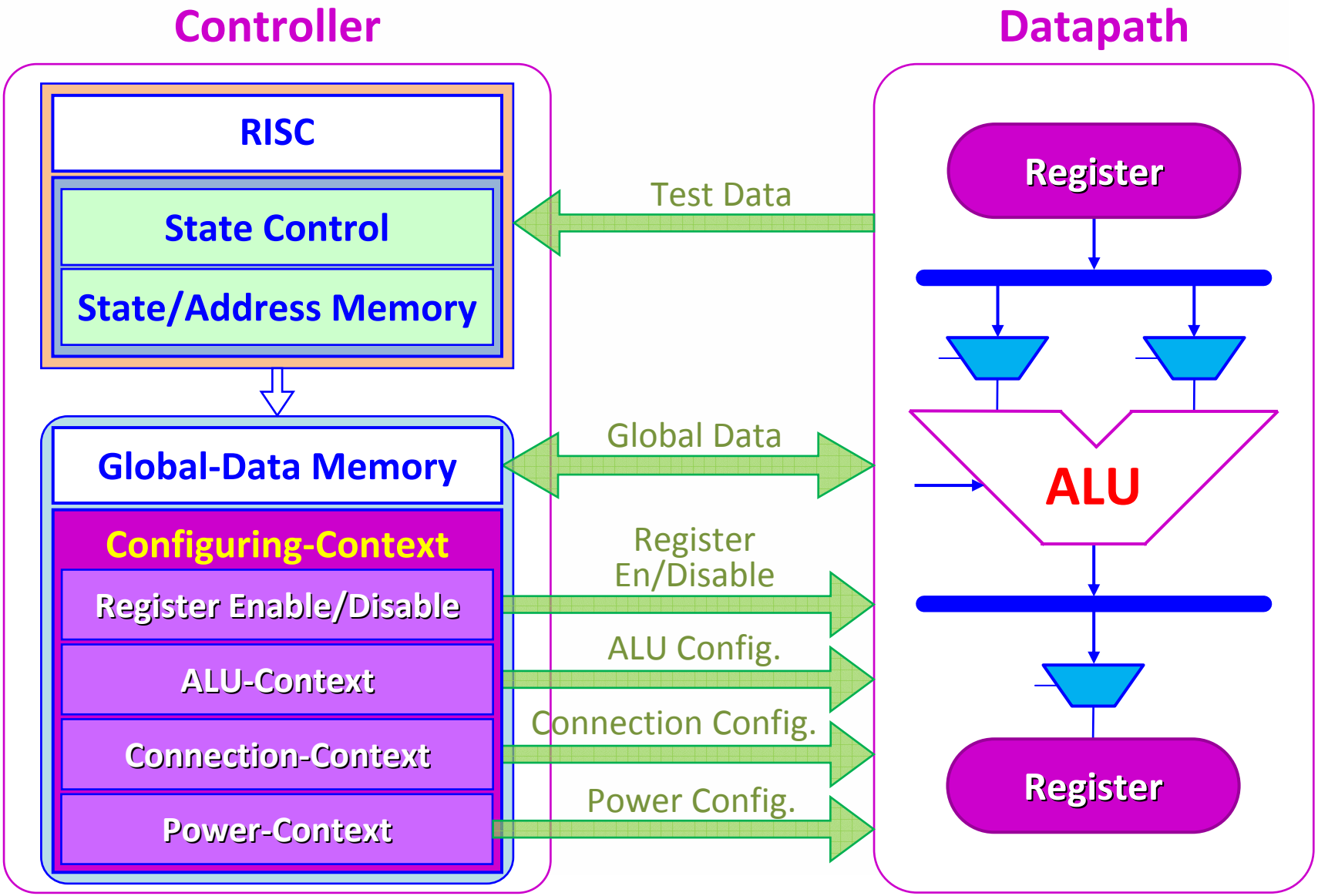
Memory Based Programmable-FSM

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

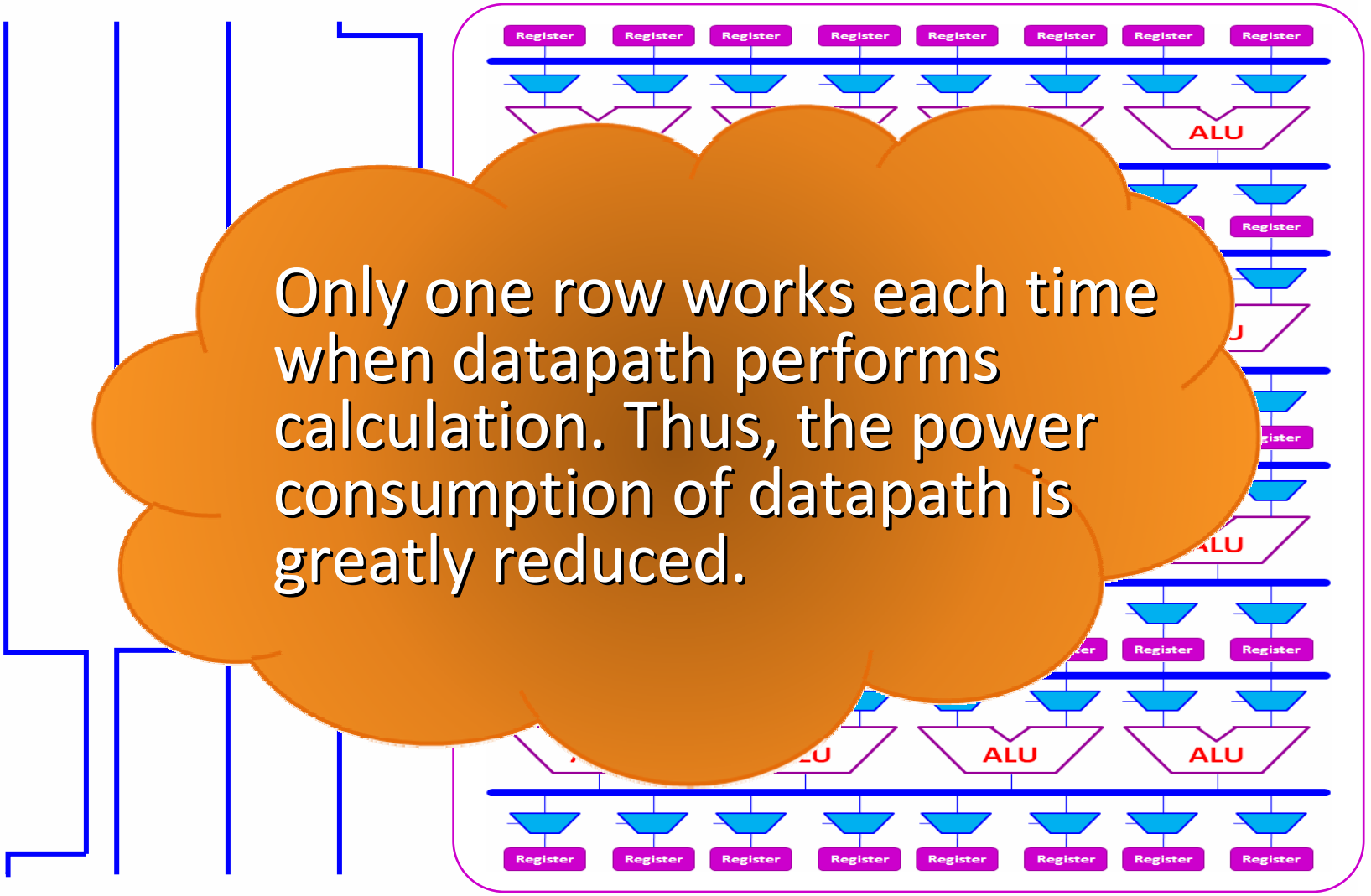




Control-Flow & Configuring Context

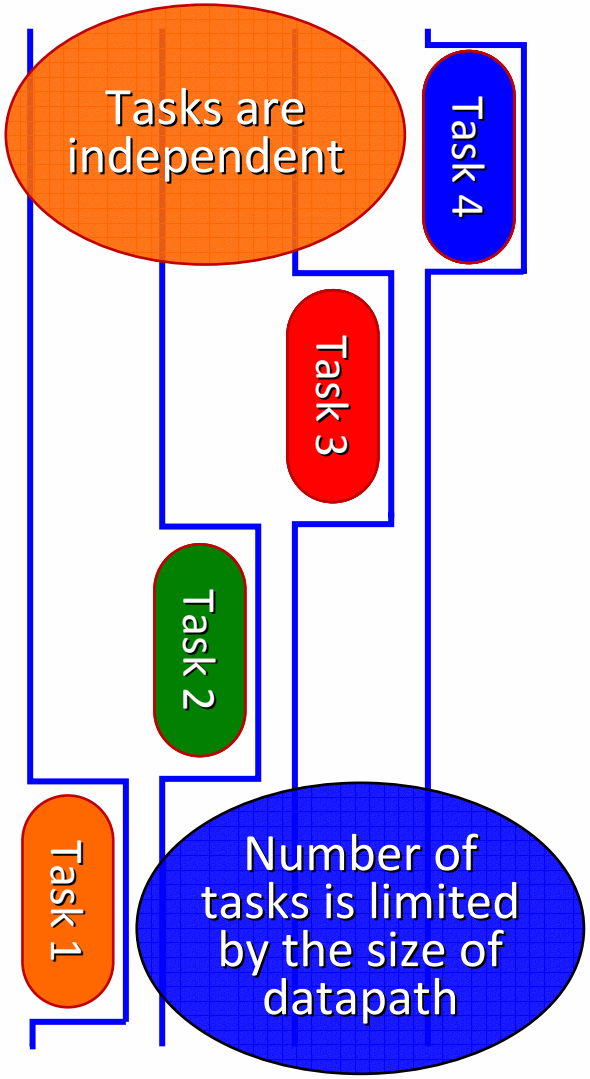


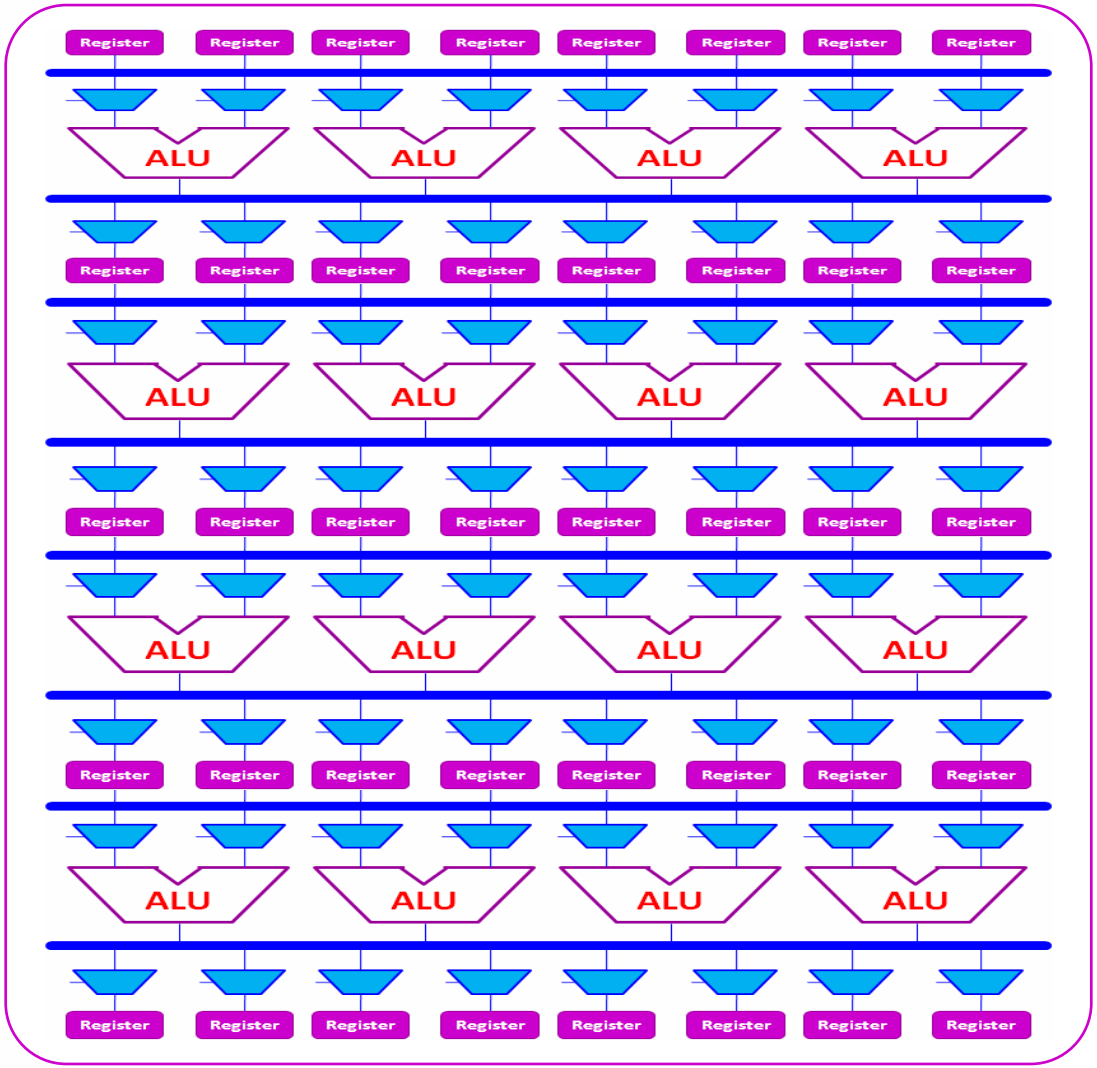
Power Consumption Reduction



Only one row works each time when datapath performs calculation. Thus, the power consumption of datapath is greatly reduced.

Multi-Task Pipeline Execution





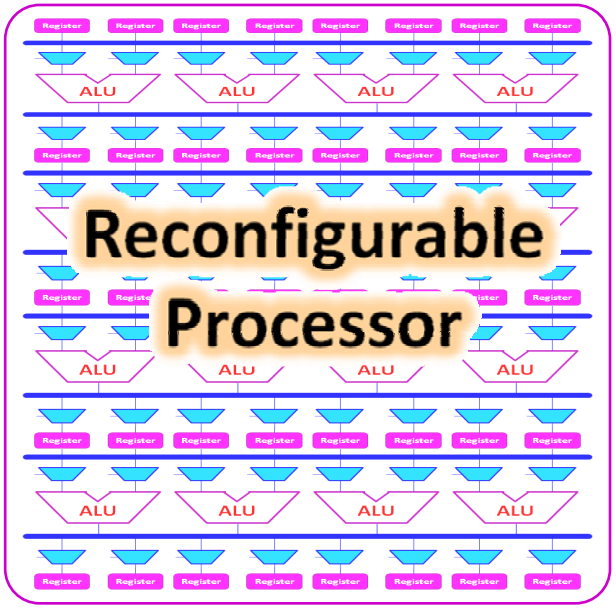
Compiler

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

```
Int main(void)
{
……
func(…, …)
……
dct(…, ,,,)
……
……
}
```

```
func(…, …)
{
……
}
```

```
dct(…, …)
{
……
}
```



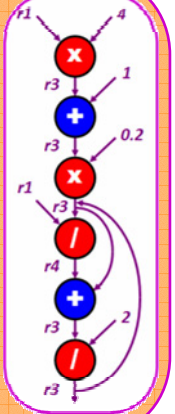
Compiler

- 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;
}
….
}
```



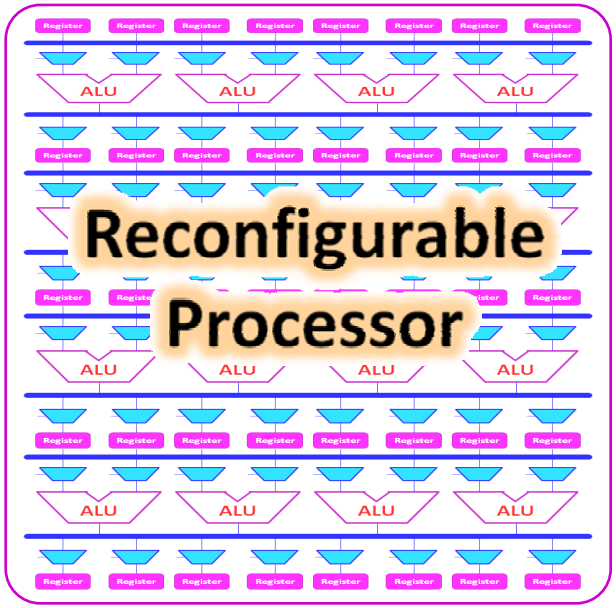
Compiler: Special Features

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

```
Int main(void)
{
……
func(…, …)
……
dct(…, ,,,)
……
……
}
```

```
func(…, …)
{
……
}
```

```
dct(…, …)
{
……
}
```



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

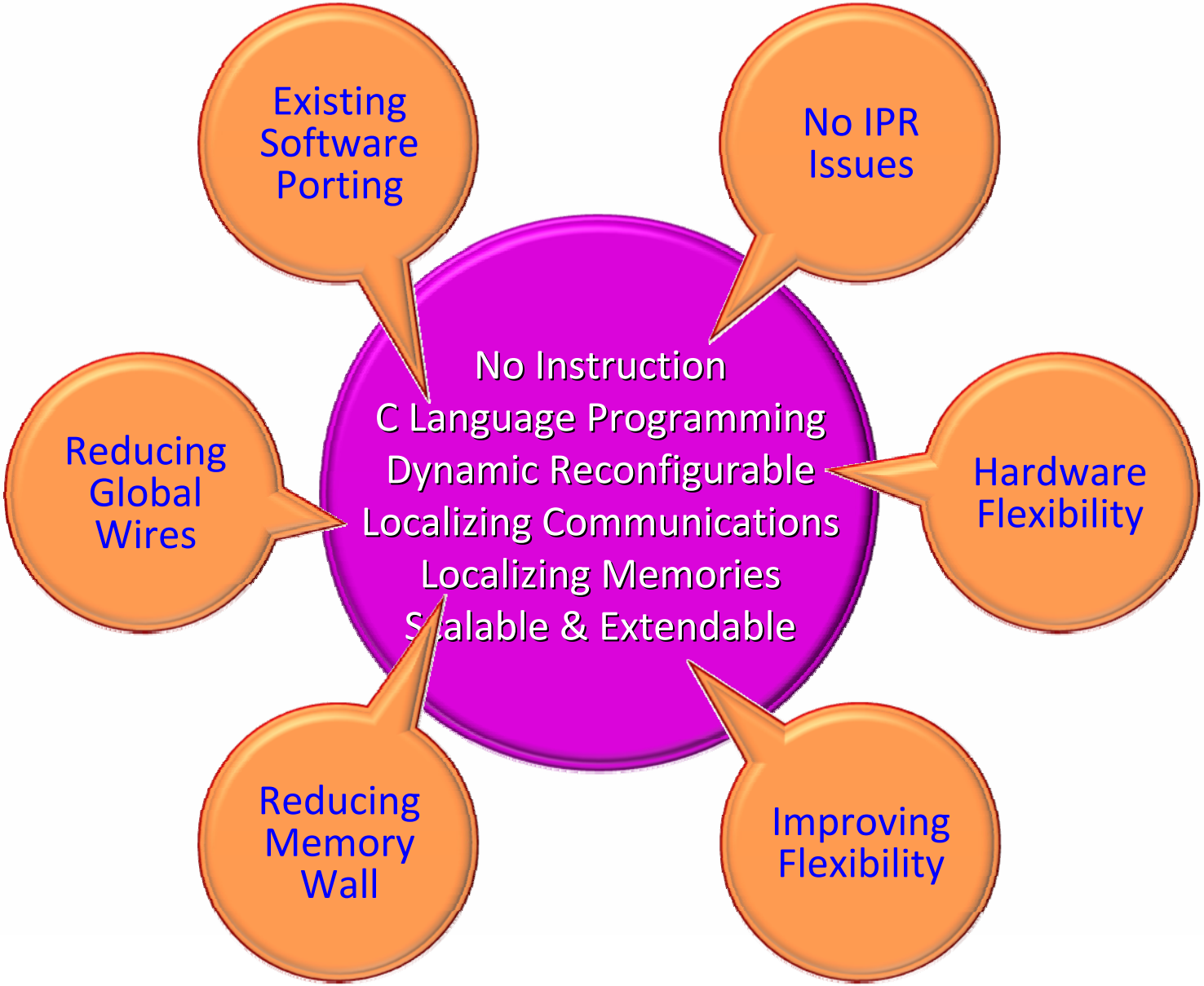
Summary

- 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.

Outlines

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

Basic Requirement



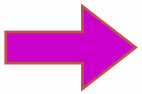
Programming Language

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

```
Int main(void)
{
……
func(…, …)
……
dct(…, ,,,)
……
……
}
```

```
func(…, …)
{
……
}
```

```
dct(…, …)
{
……
}
```



Compiler

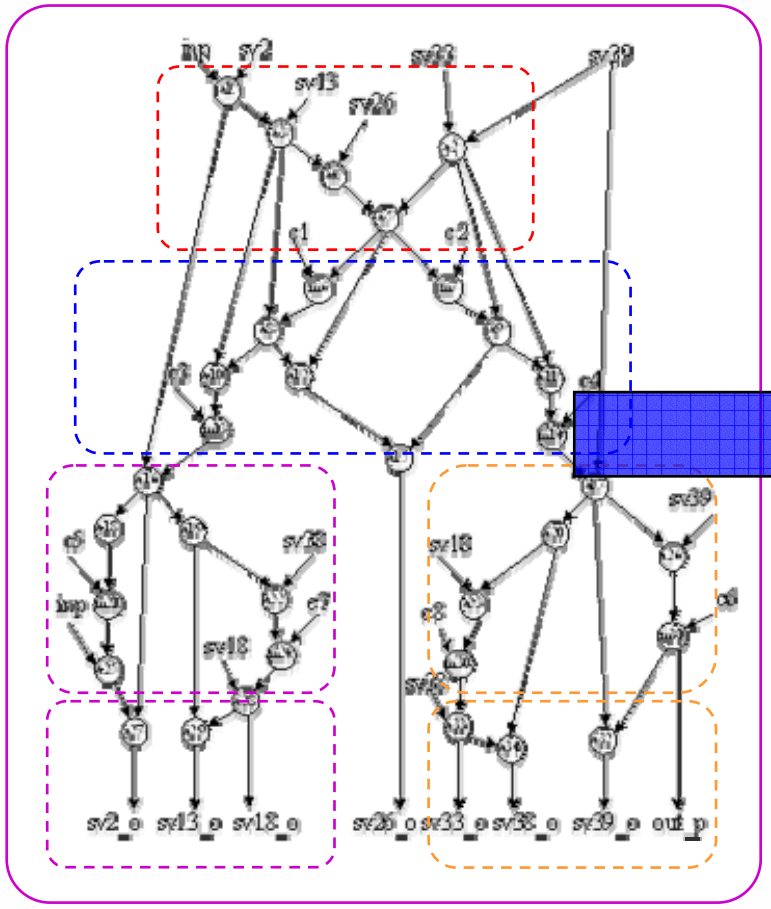
- Syntax Check
- Code Profiling
- Code Transformation
- Code Optimization
- Data-Flow Generation
- Task Partitioning
- Task Scheduling
- Allocation
- Connection Scheme
- Mappings
- Evaluations
- Context Generation

Challenges

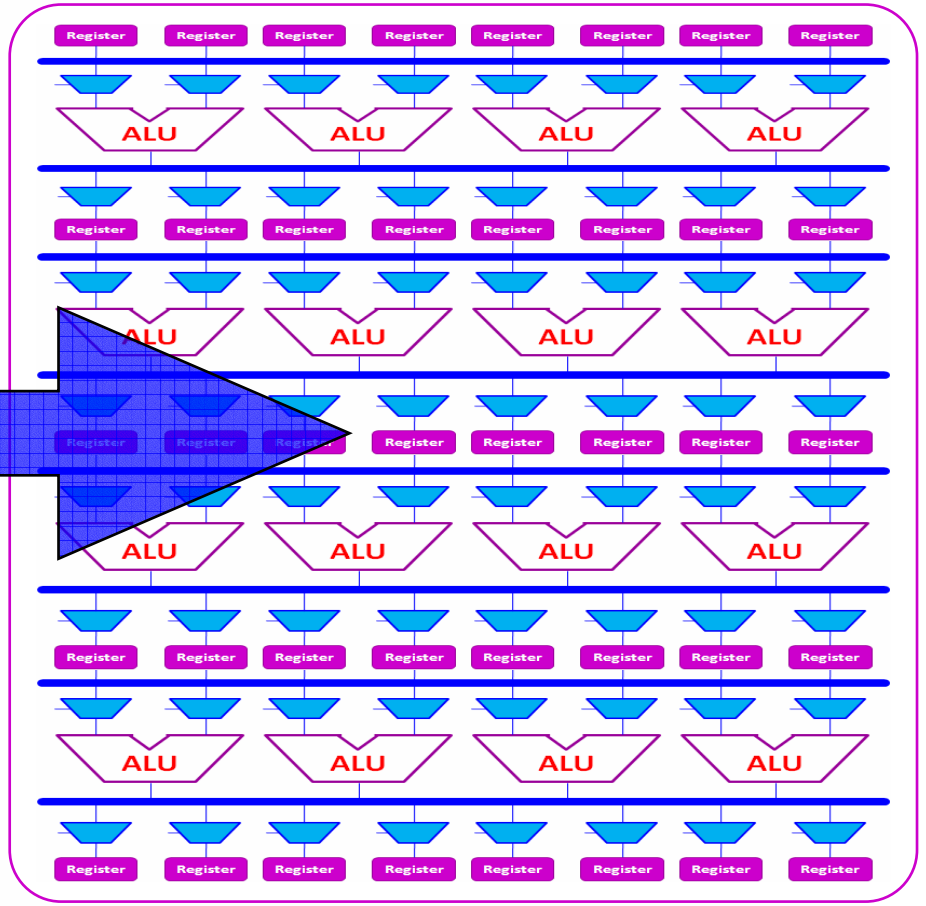
- Available software
- C language
- Pointers
- Loops
- Recursive call
- Vector
- matrix
- ….

Dimension Limited: Task Mapping

Task Partitioning
Task Flow Generation
Task Dependency

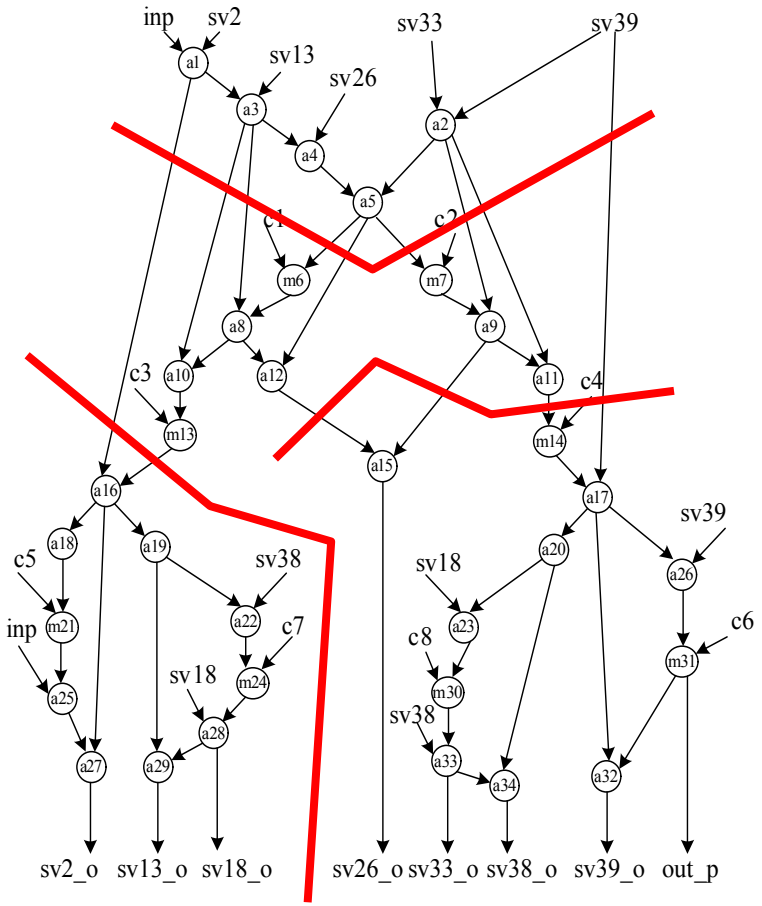


Task Scheduling
Datapath Allocation
Mapping Scheme

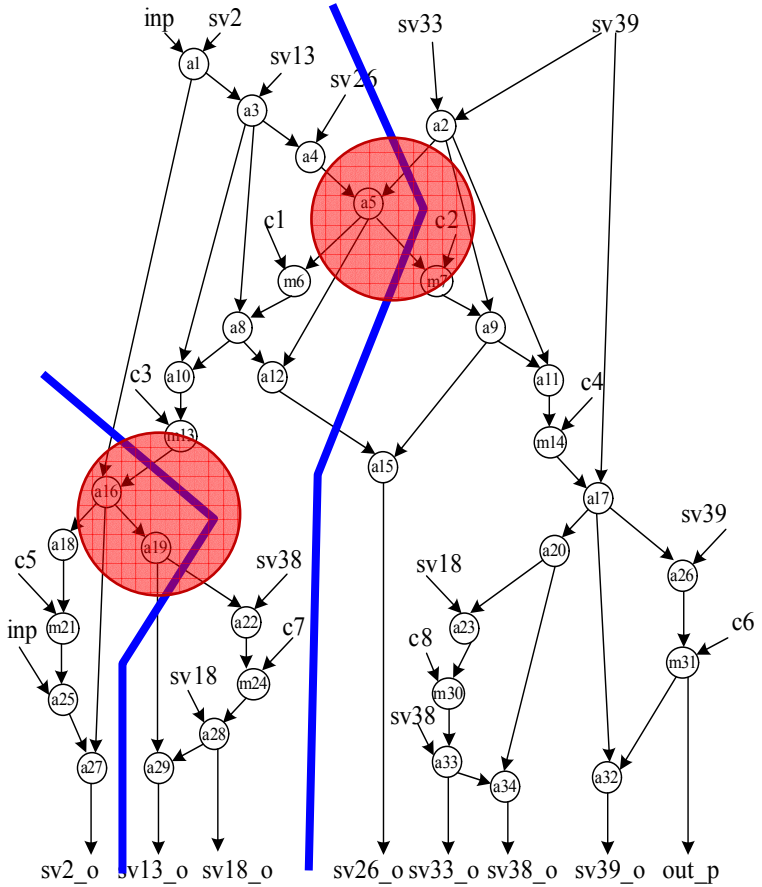


Partitioning and Deadlock

Task Graph Partitioning (No Deadlock)

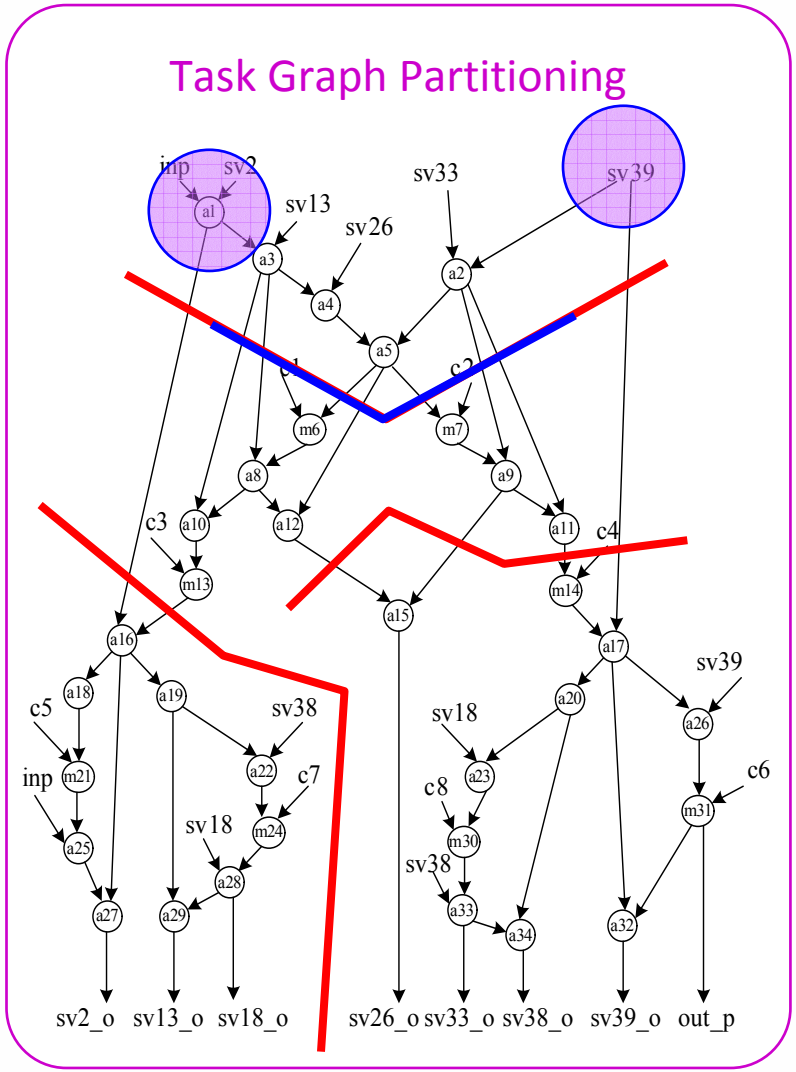


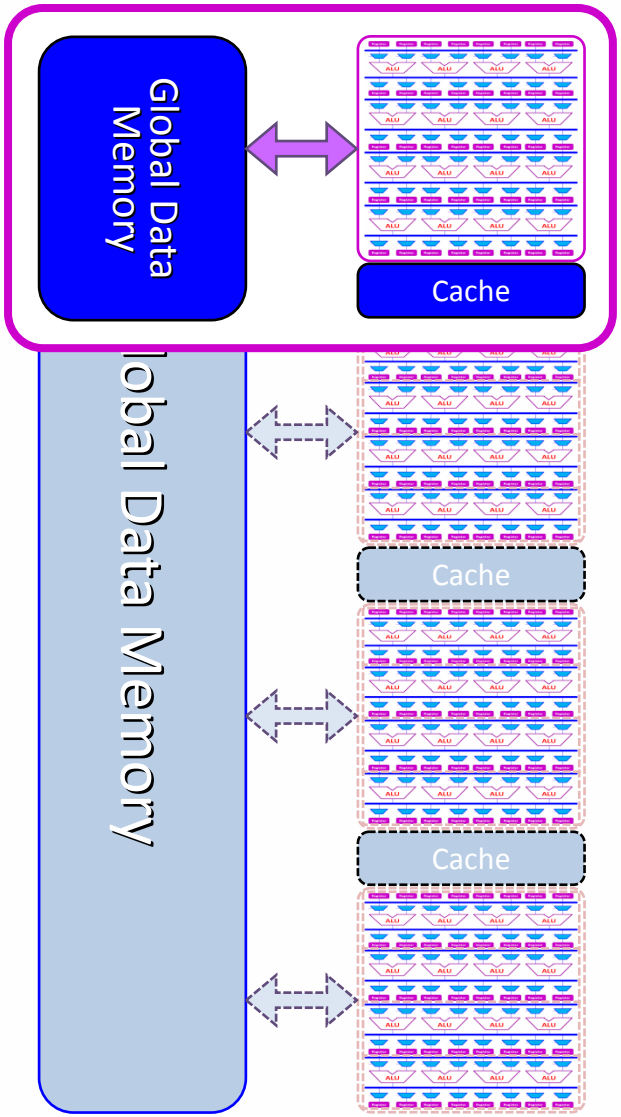
Task Graph Partitioning (Deadlock)



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

Global Data Passing





Operating System

PROGRAM A

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

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

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

PROGRAM B

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

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

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

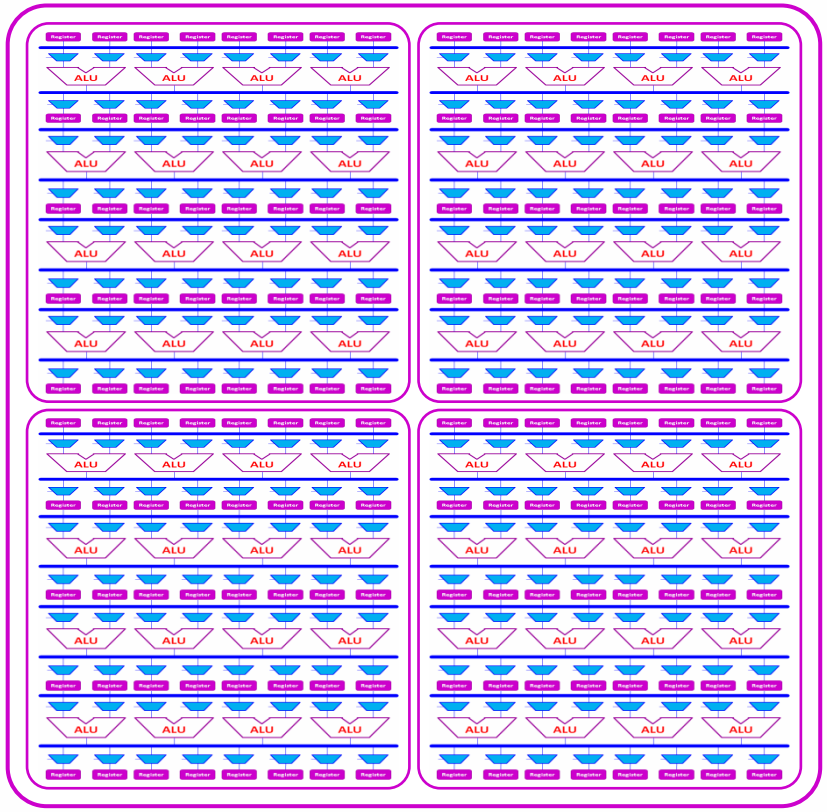
PROGRAM C

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

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

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

- Multi-Task management
- Resource management
- Time sharing and resource sharing



- Multi-Controller
- Dynamic Resource allocation
- Queue-up

Summary

- 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.

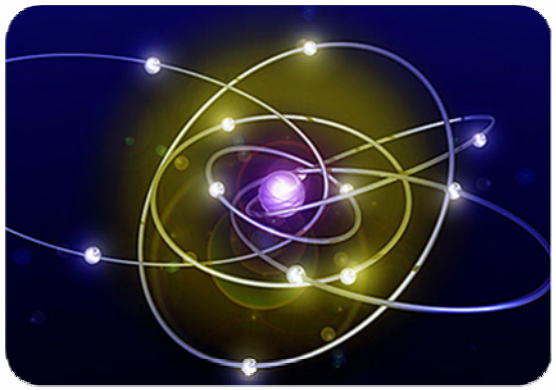
Outlines

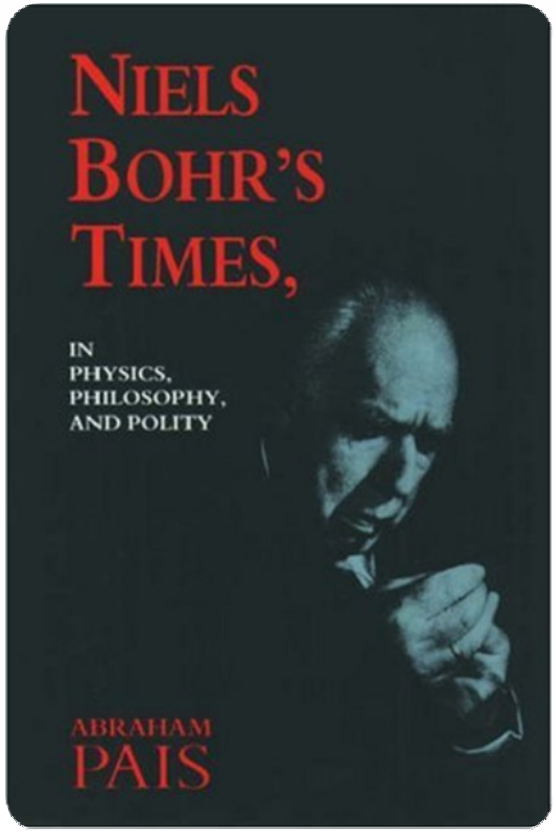
- Introduction
- Von Neumann Architecture
- Reconfigurable Computing
- Major Challenges
- Conclusions

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.

Predicting the Future



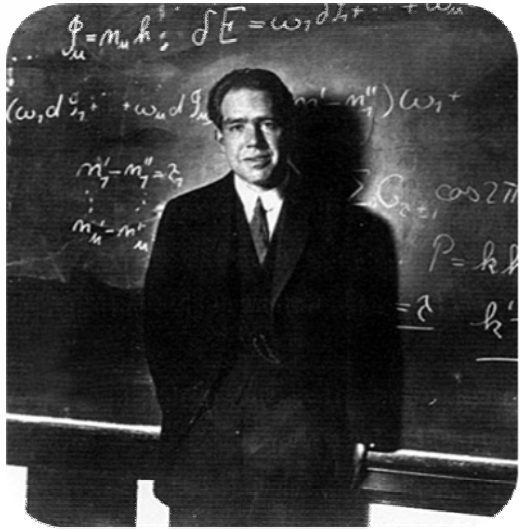


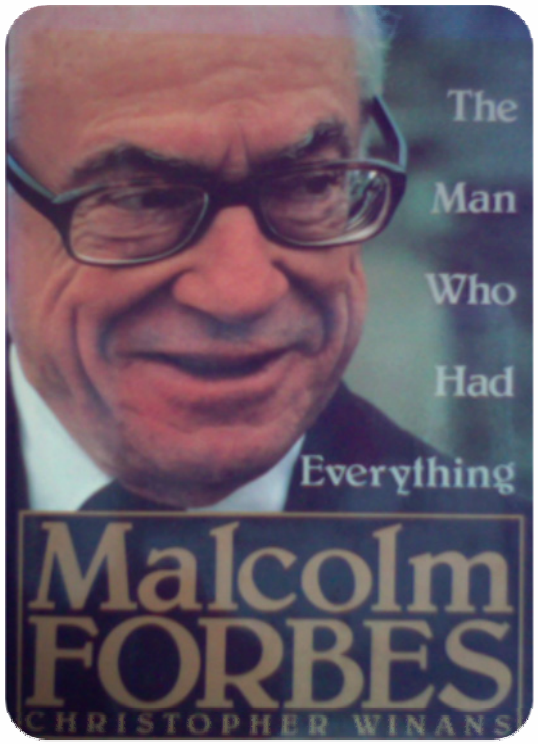
Prediction is very difficult, especially if it's about the future.

- 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。

Wally Rhines, Chairman & CEO, Mentor Graphics, August 2010



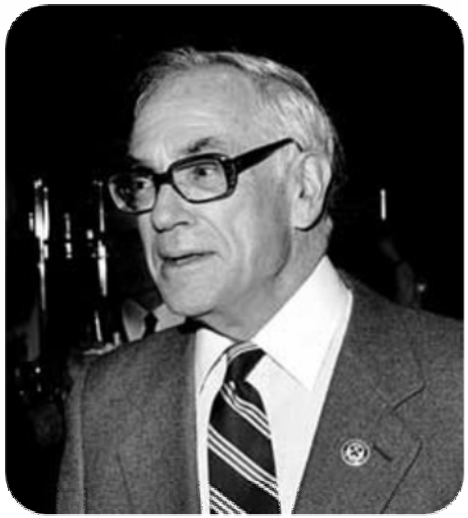


Anyone who says businessmen deal in facts—not fiction—has never read old five-year projections…"

- 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