Evolution of Microprocessors

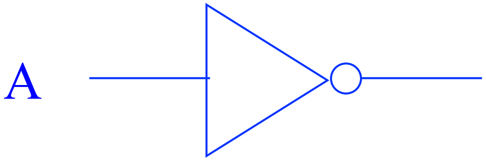
Olivier Temam
INRIA Saclay

What Is The Goal of A Microprocessor ?

- Executing any algorithm:
  - Computations
  - Evaluating logic expressions
  - Storing data

```
     A=0
Label:
     A=A+1
     if (A<5) goto Label
```

This is All We Need

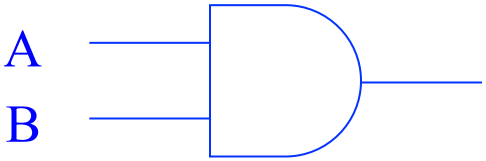


$Z = A'$

NOT

A	Z
0	**1**
1	**0**

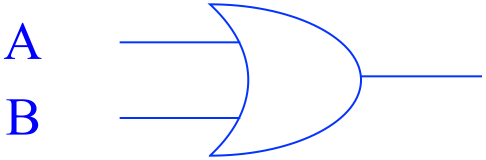
0: false
1: true



$Z = A . B$

AND

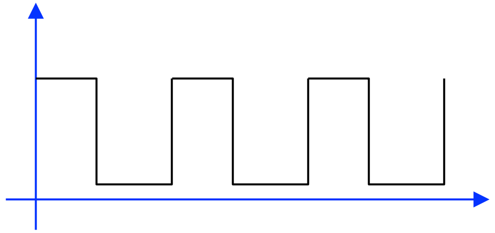
A	B	Z
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**



$Z = A + B$

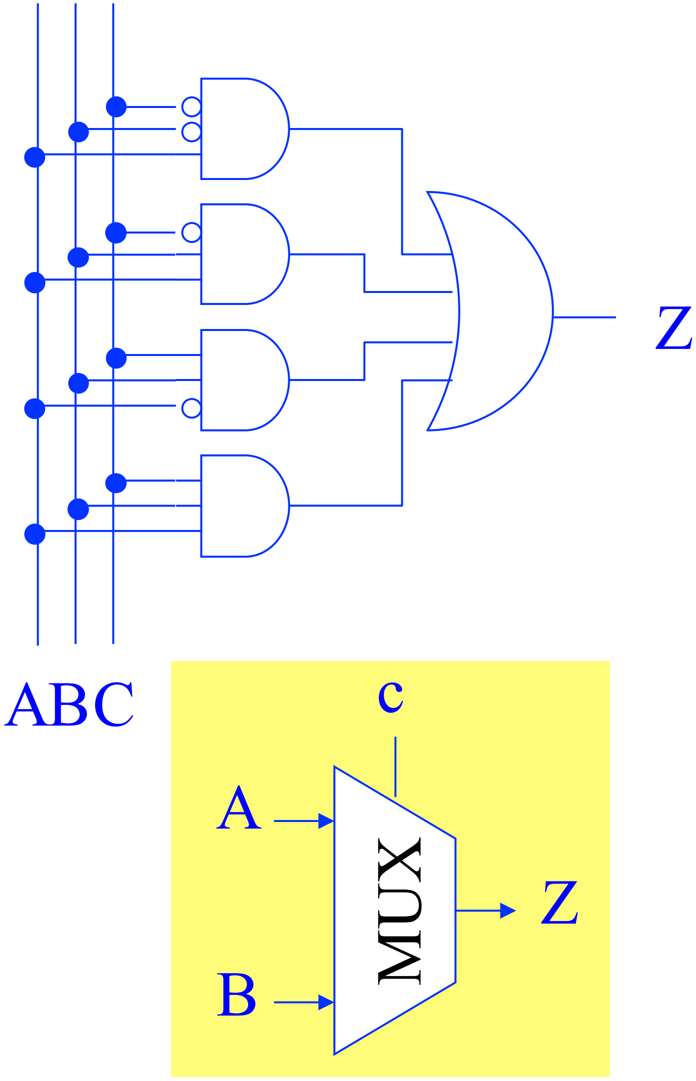
OR

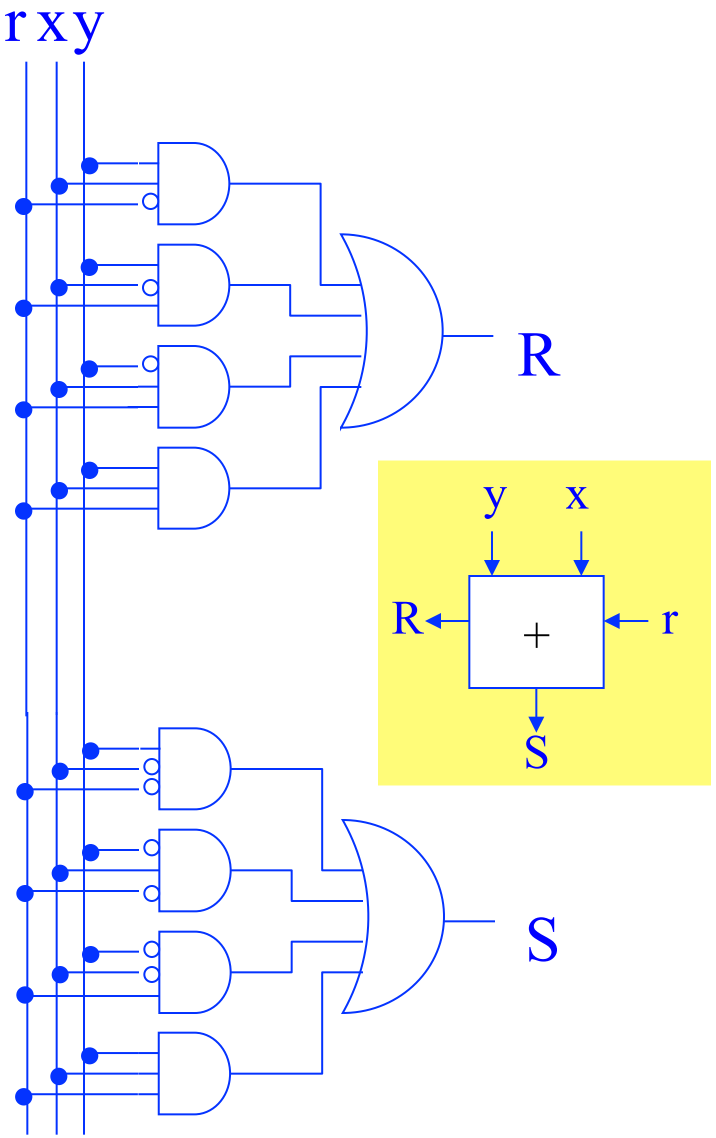
A	B	Z
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**



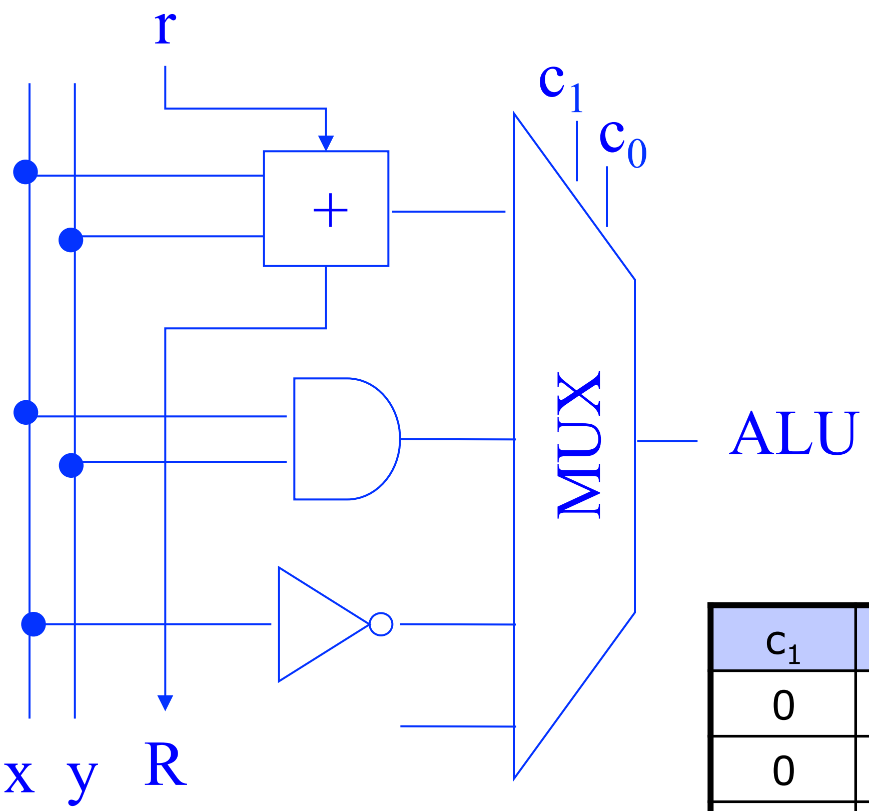
- Can build any logic function (Shannon)

Logic & Computational Circuits

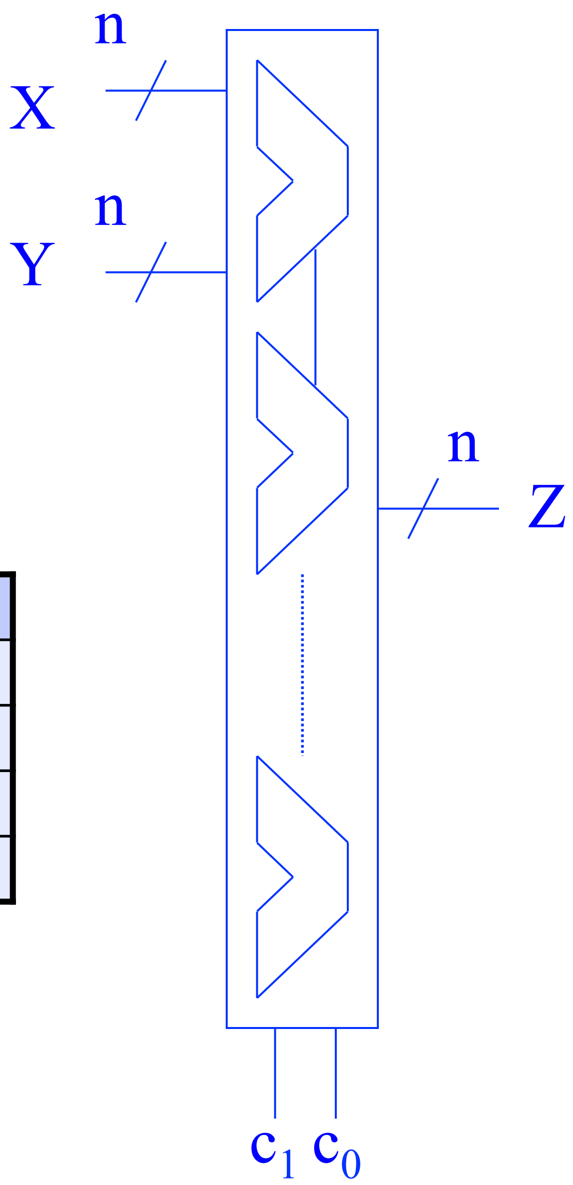




The Heart of the Processor: ALU

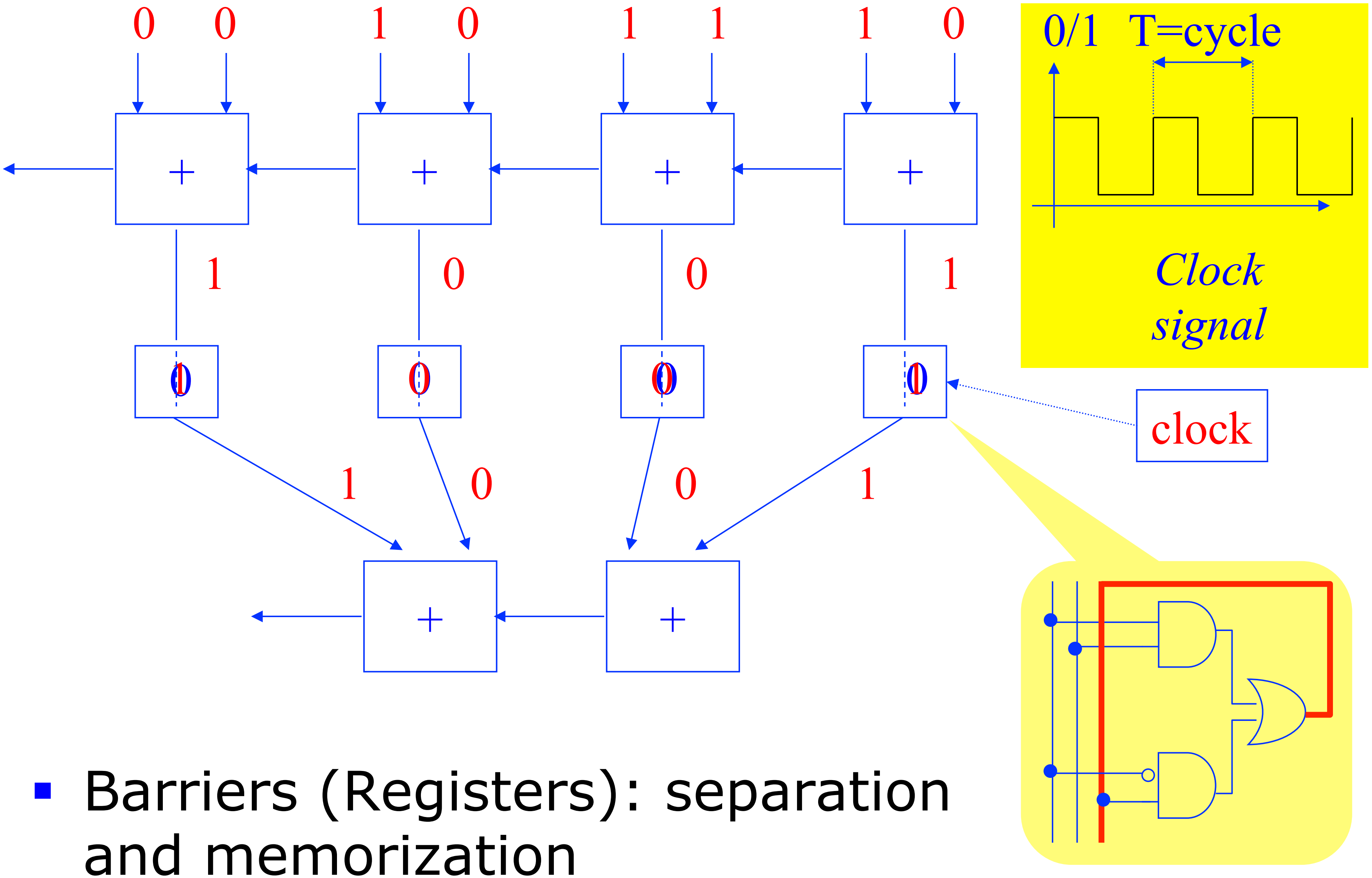


c_1	c_0	**ALU**
0	0	**ADD**
0	1	**AND**
1	0	**NOT**
1	1	***d***



- ALU: Arithmetic and Logic Unit
- Embryo of instruction set:
    - ADD, AND, NOT

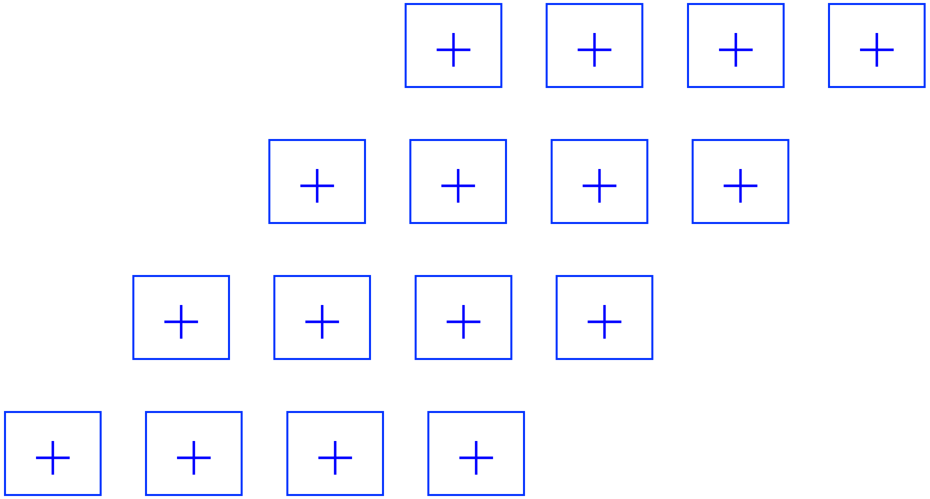
Connecting Several Operators



- Barriers (Registers): separation and memorization

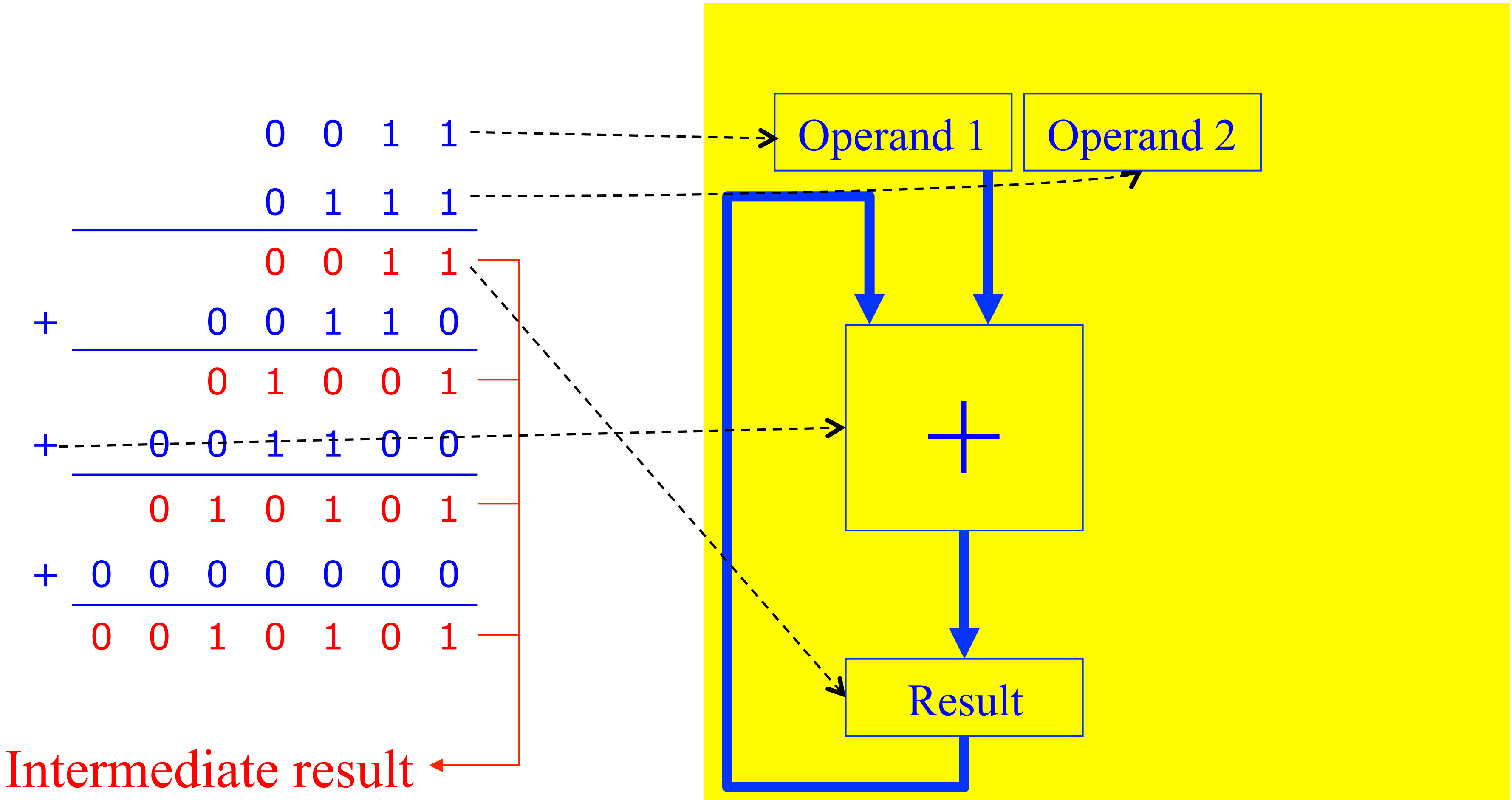
Complex Circuits

```
          0 0 1 1
          0 1 1 1
    -------------
          0 0 1 1
+       0 0 1 1
+     0 0 1 1
+   0 0 0 0
    -------------
    0 0 1 0 1 0 1
```



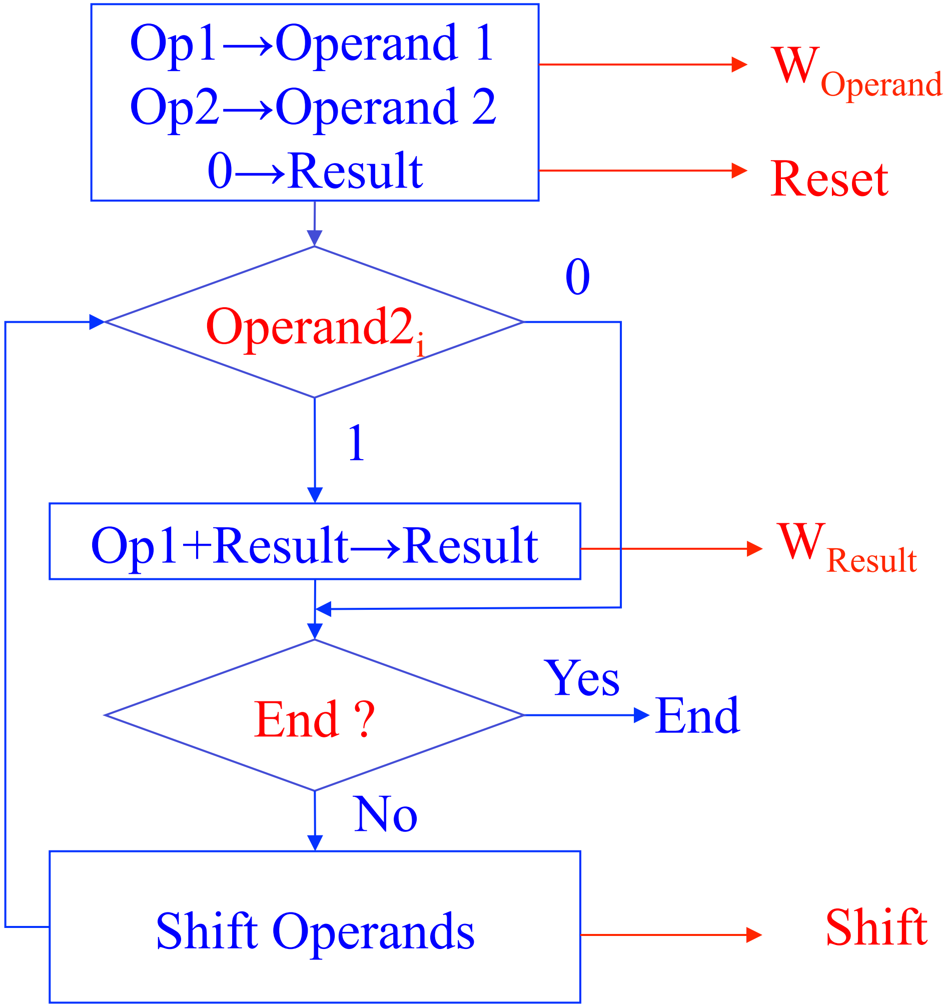
- Multiplier as a function

Complex Circuits



- Multiplier as a sequence of operations

Controlling Complex Circuits



Op1→Operand 1
Op2→Operand 2
0→Result

$W_{Operand}$

Reset

$Operand2_i$

0

1

Op1+Result→Result

W_{Result}

End ?

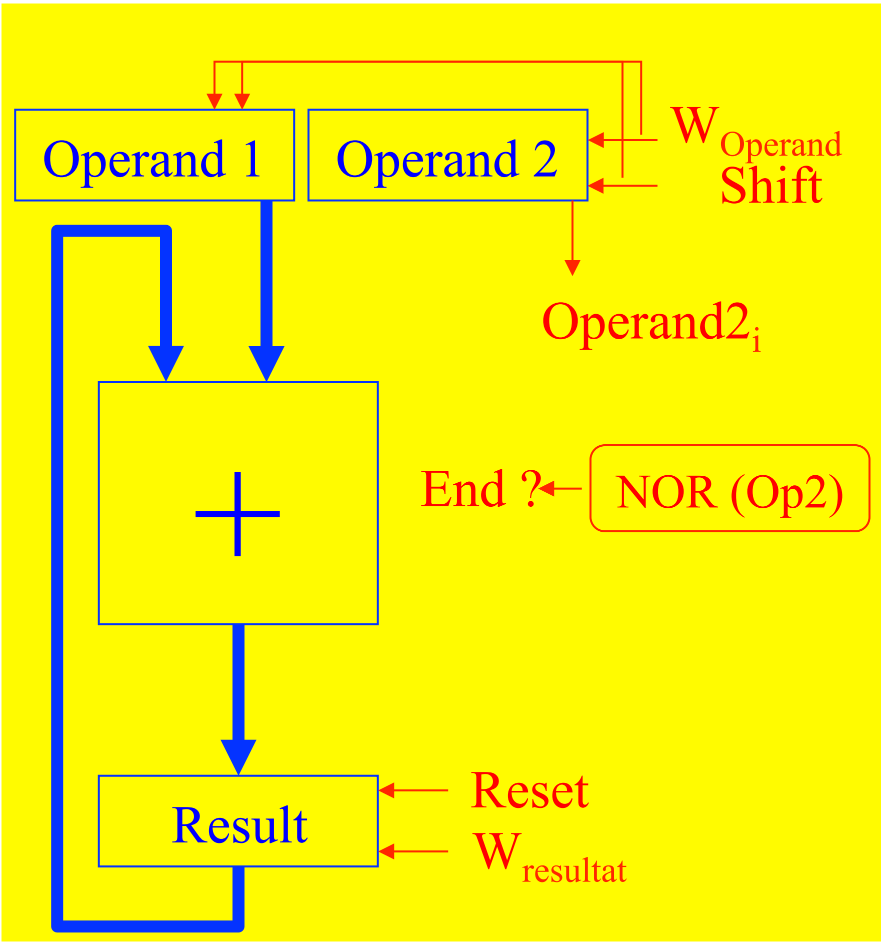
Yes

End

No

Shift Operands

Shift



Operand 1

Operand 2

$W_{Operand}$
Shift

$Operand2_i$

+

End ?

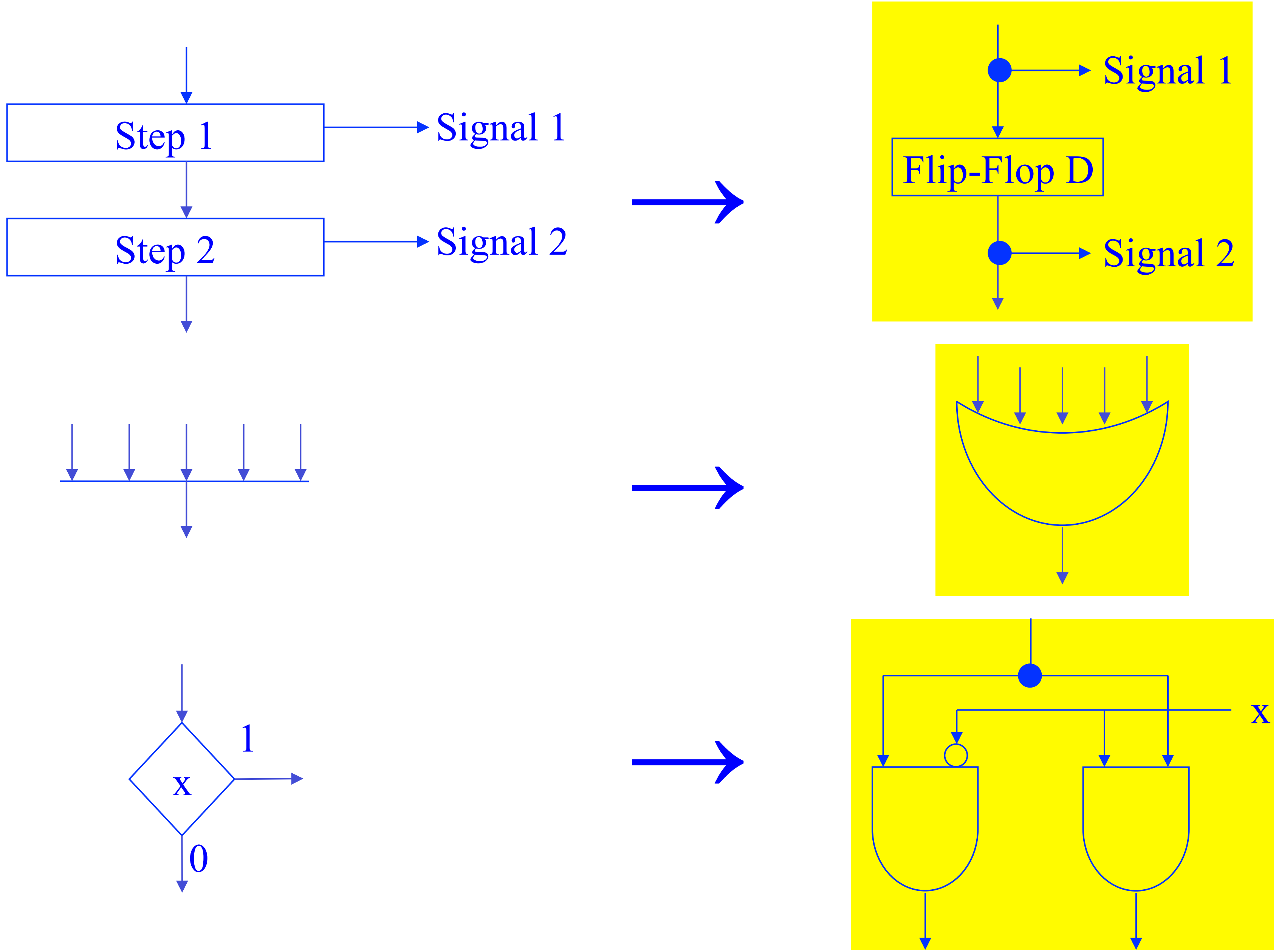
NOR (Op2)

Result

Reset

$W_{resultat}$

Converting an Algorithm into a Circuit



Step 1

Signal 1

Step 2

Signal 2

Signal 1

Flip-Flop D

Signal 2

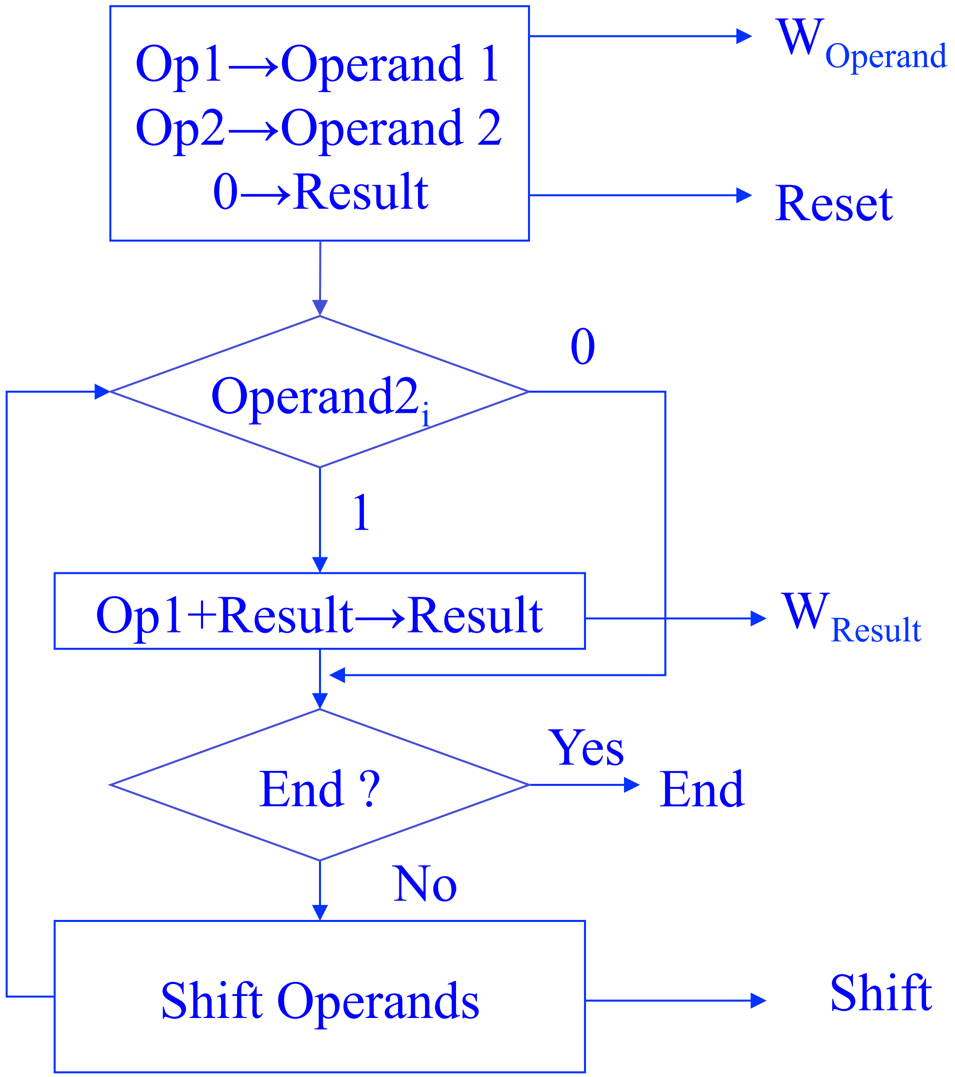
x

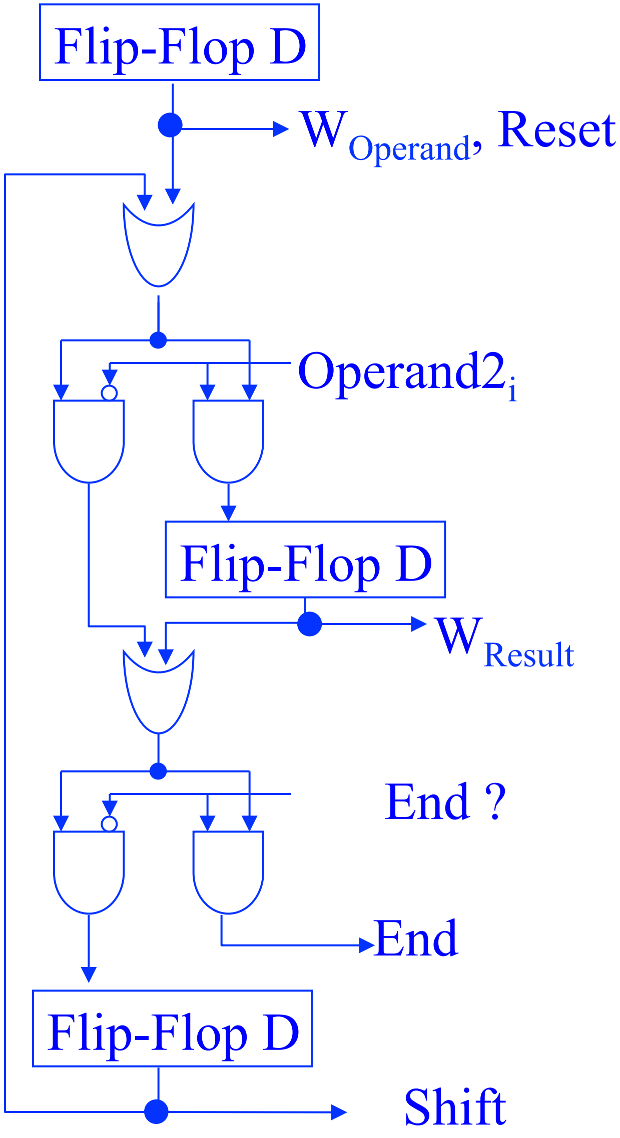
1

0

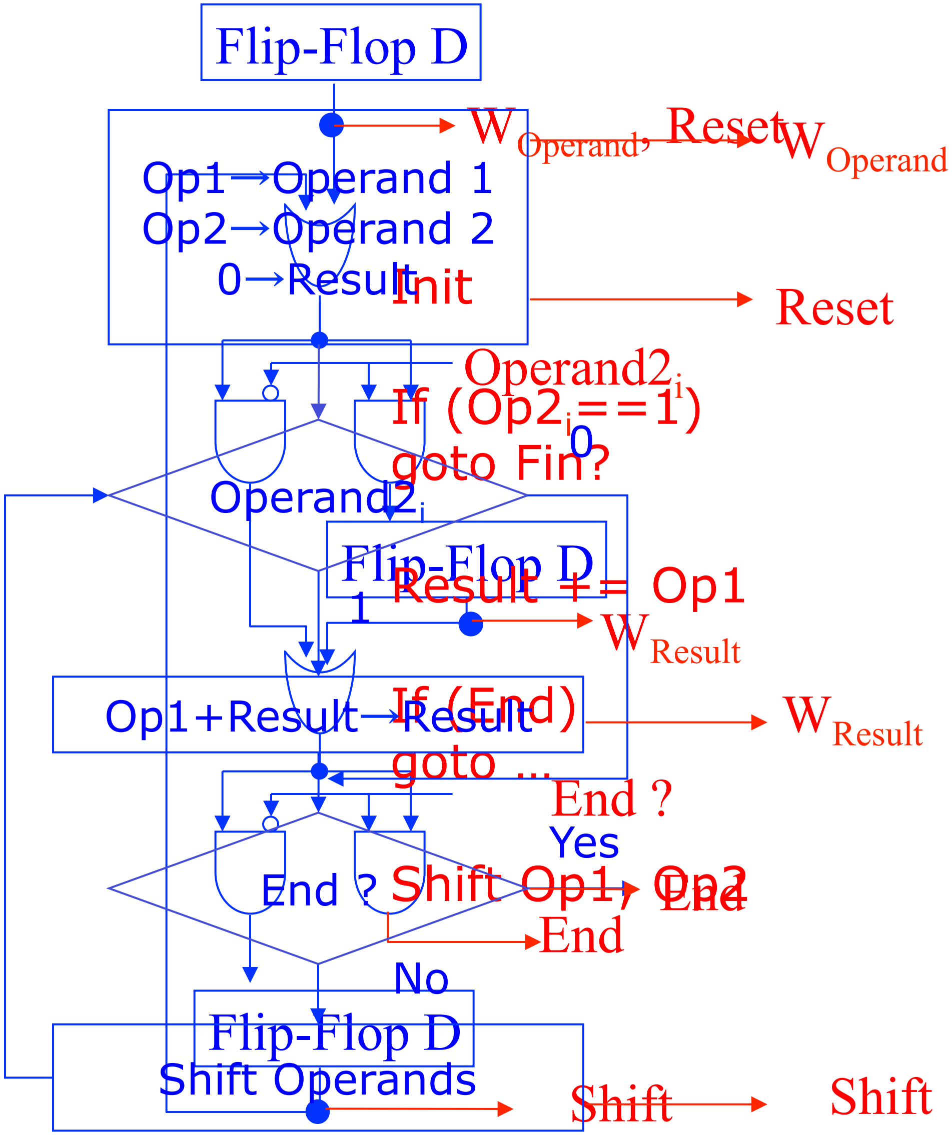
x

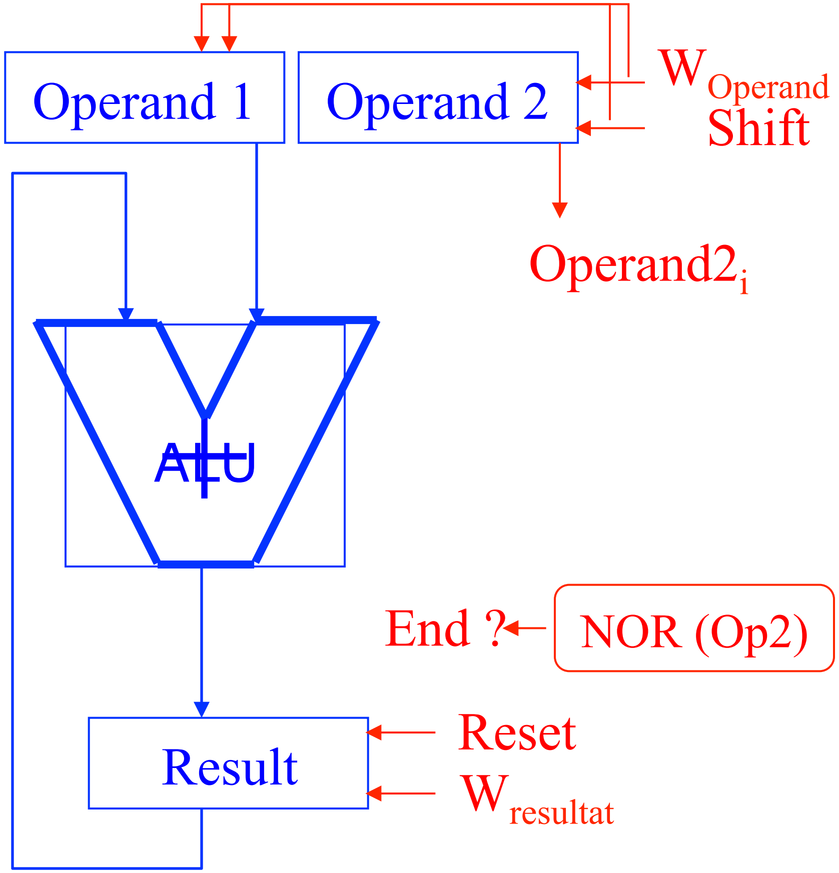
Multiplier Control Circuit





Generalization ?





Generalization ?

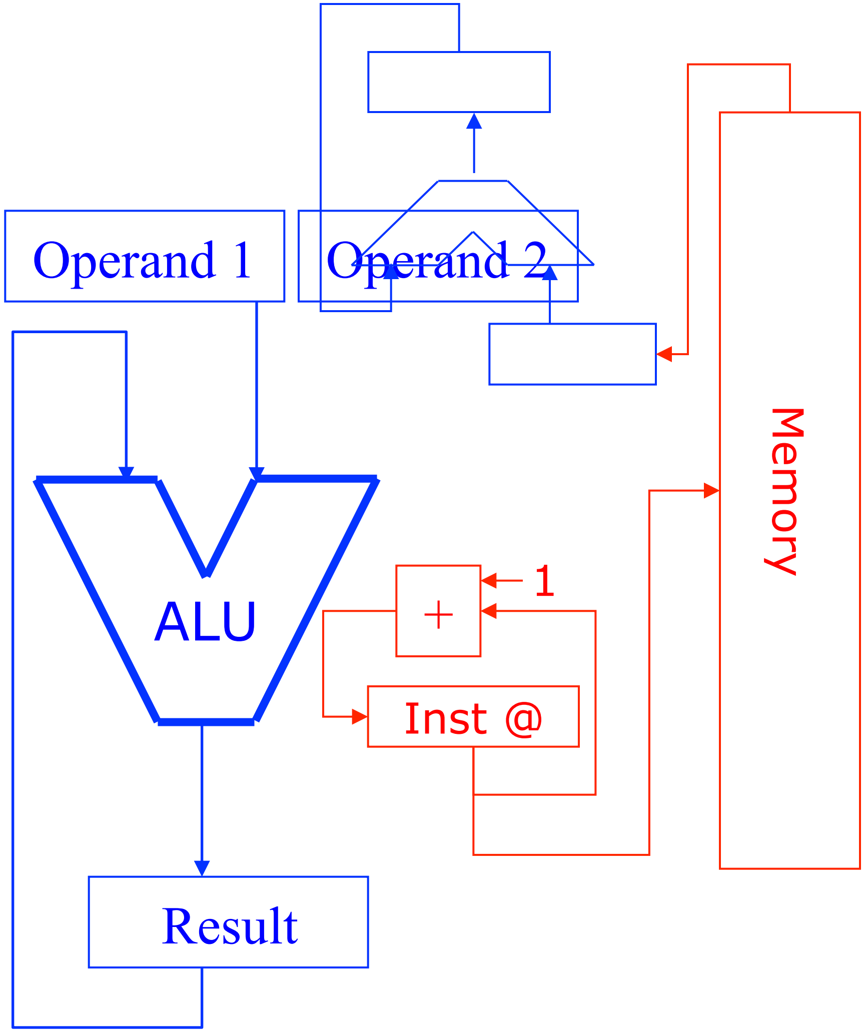
Init

If ($Op2_i$==1)
goto Fin?

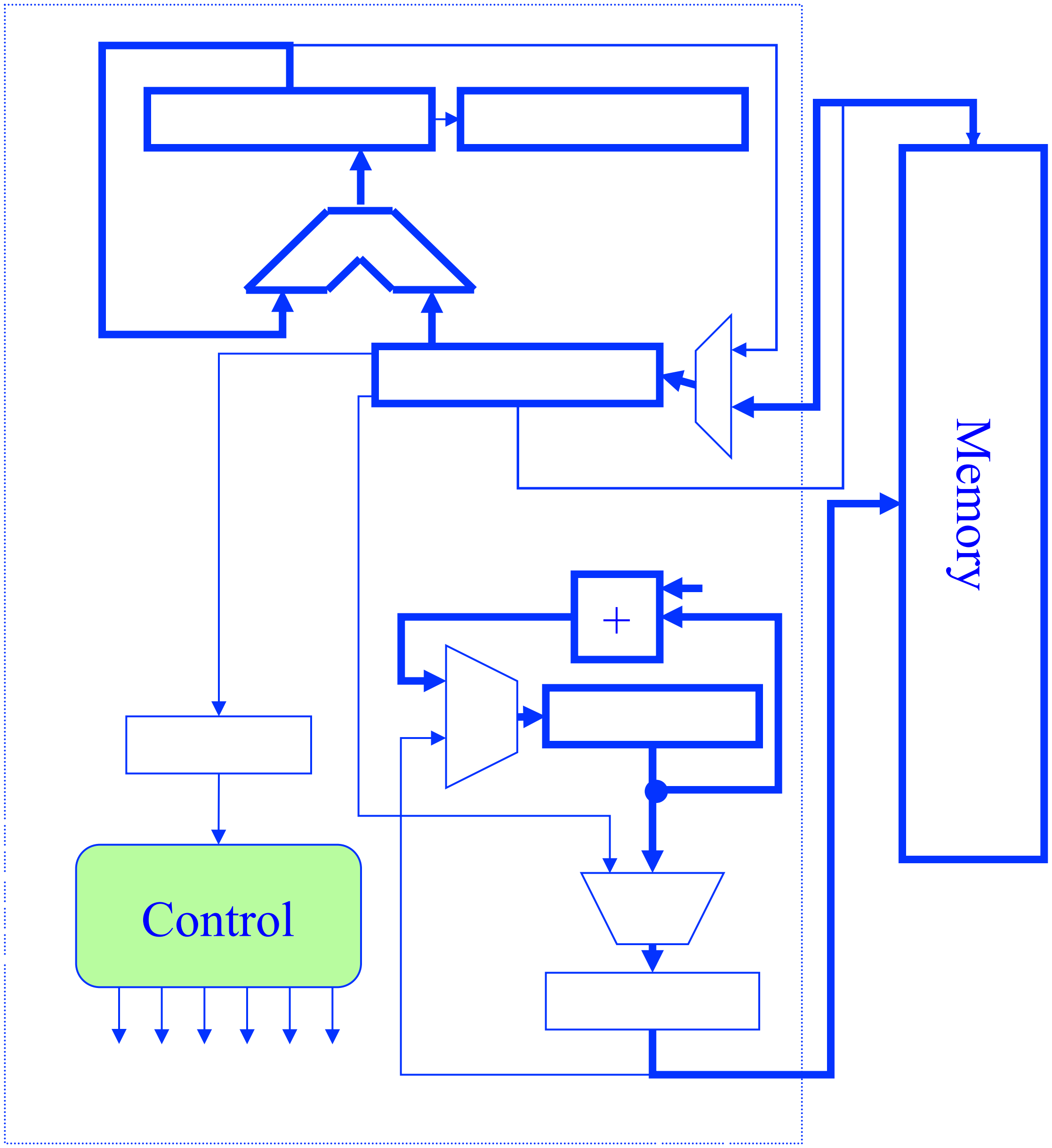
Result += Op1

If (End)
goto …

Shift Op1, Op2



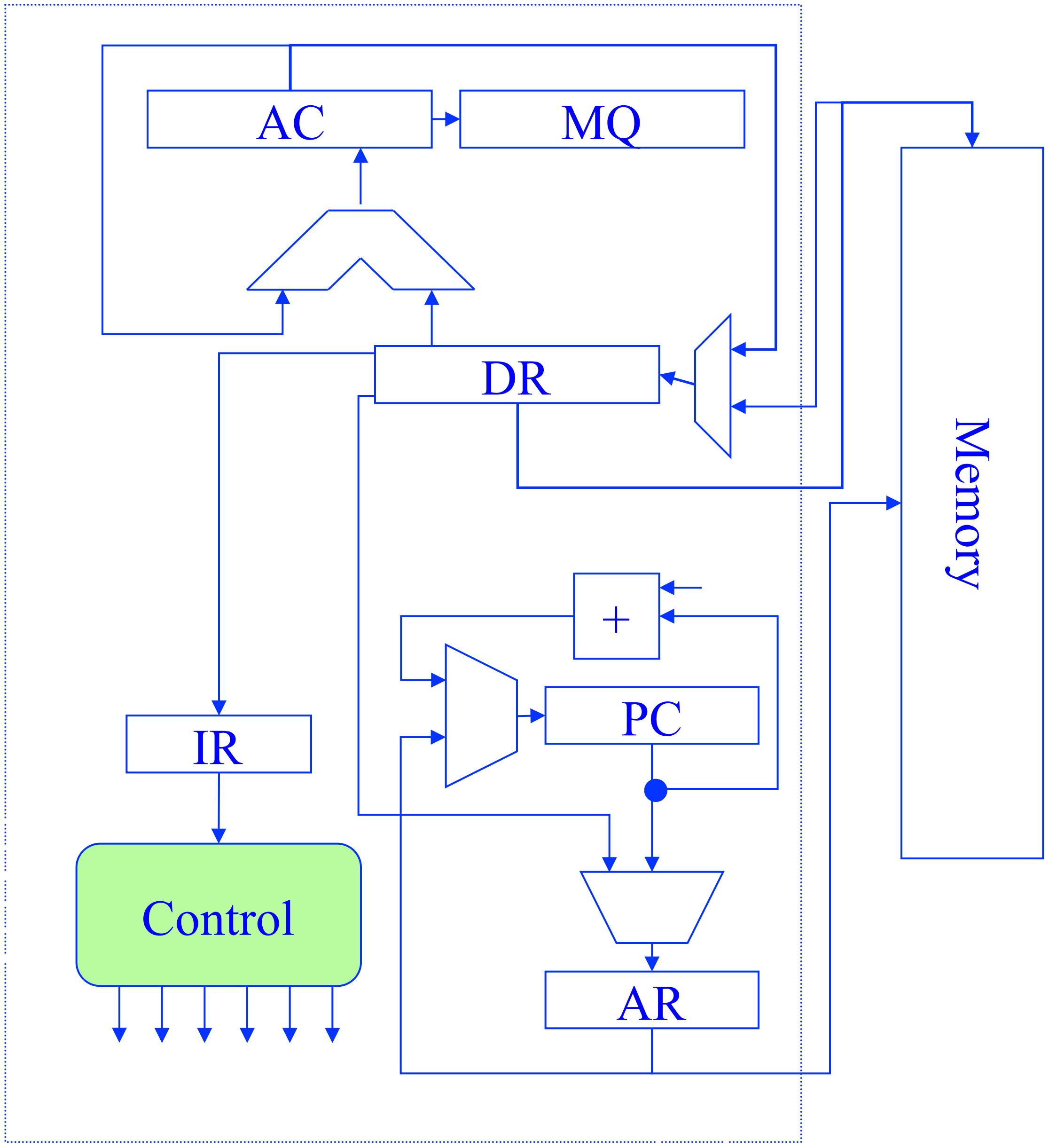
IAS - 1946



IAS - 1946

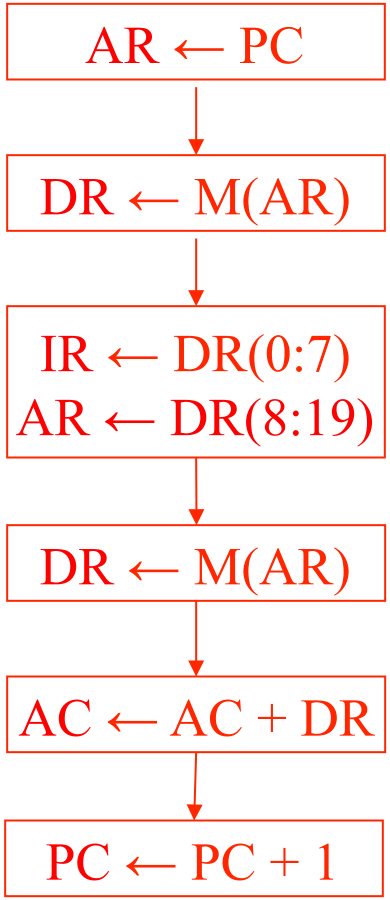
Opcode	*Operand*

Instruction



IAS - 1946

AC ← AC+M(X)



AR ← PC

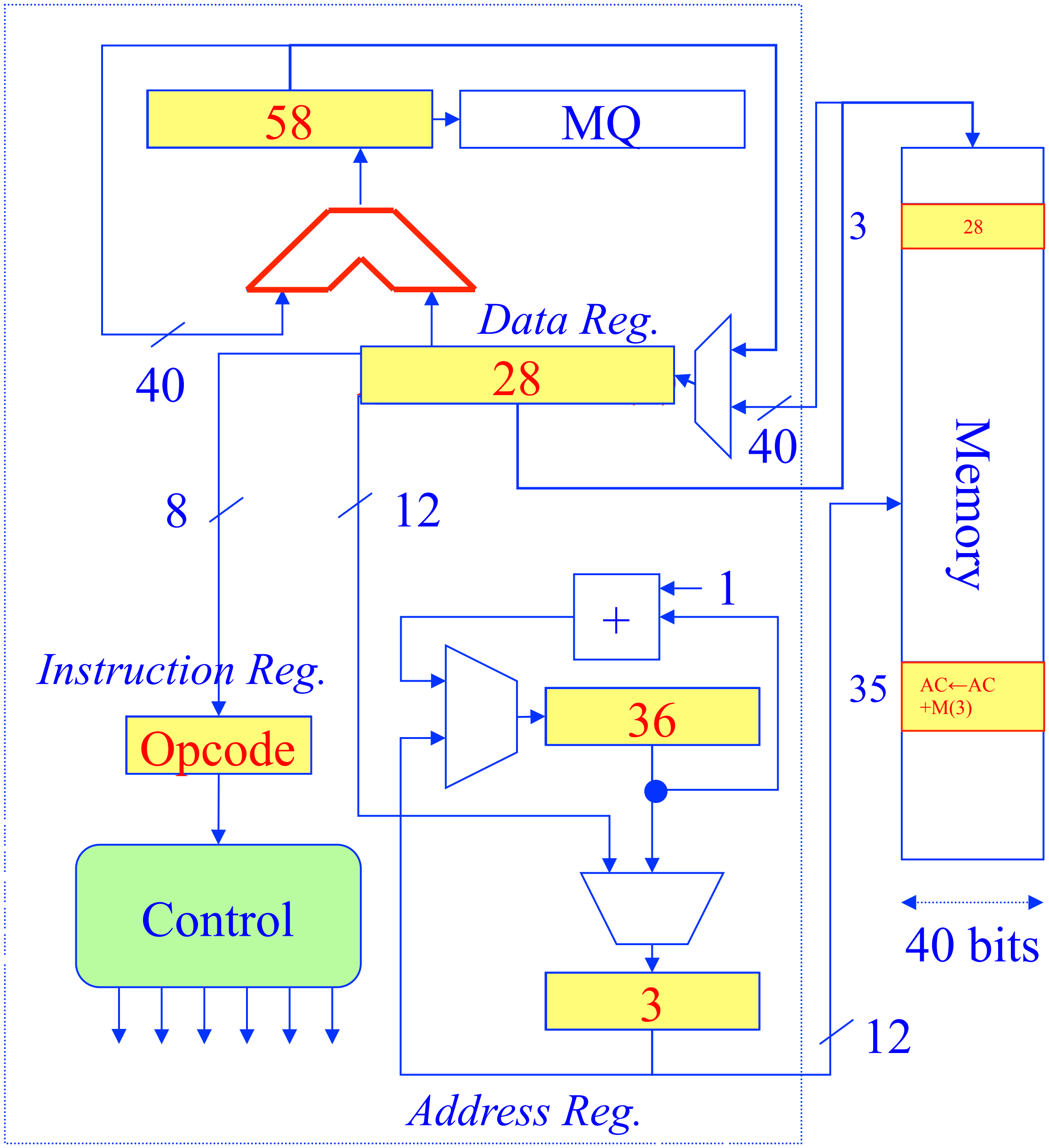
DR ← M(AR)

IR ← DR(0:7)
AR ← DR(8:19)

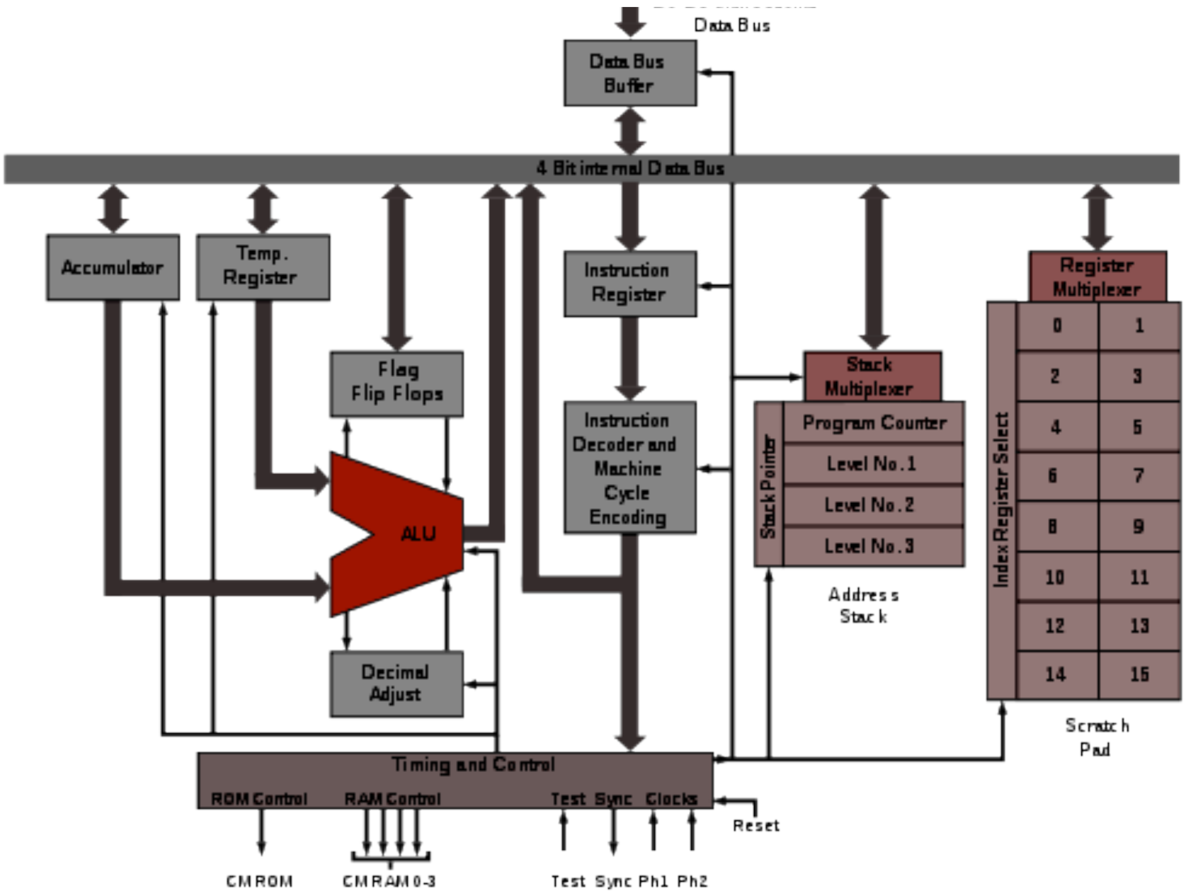
DR ← M(AR)

AC ← AC + DR

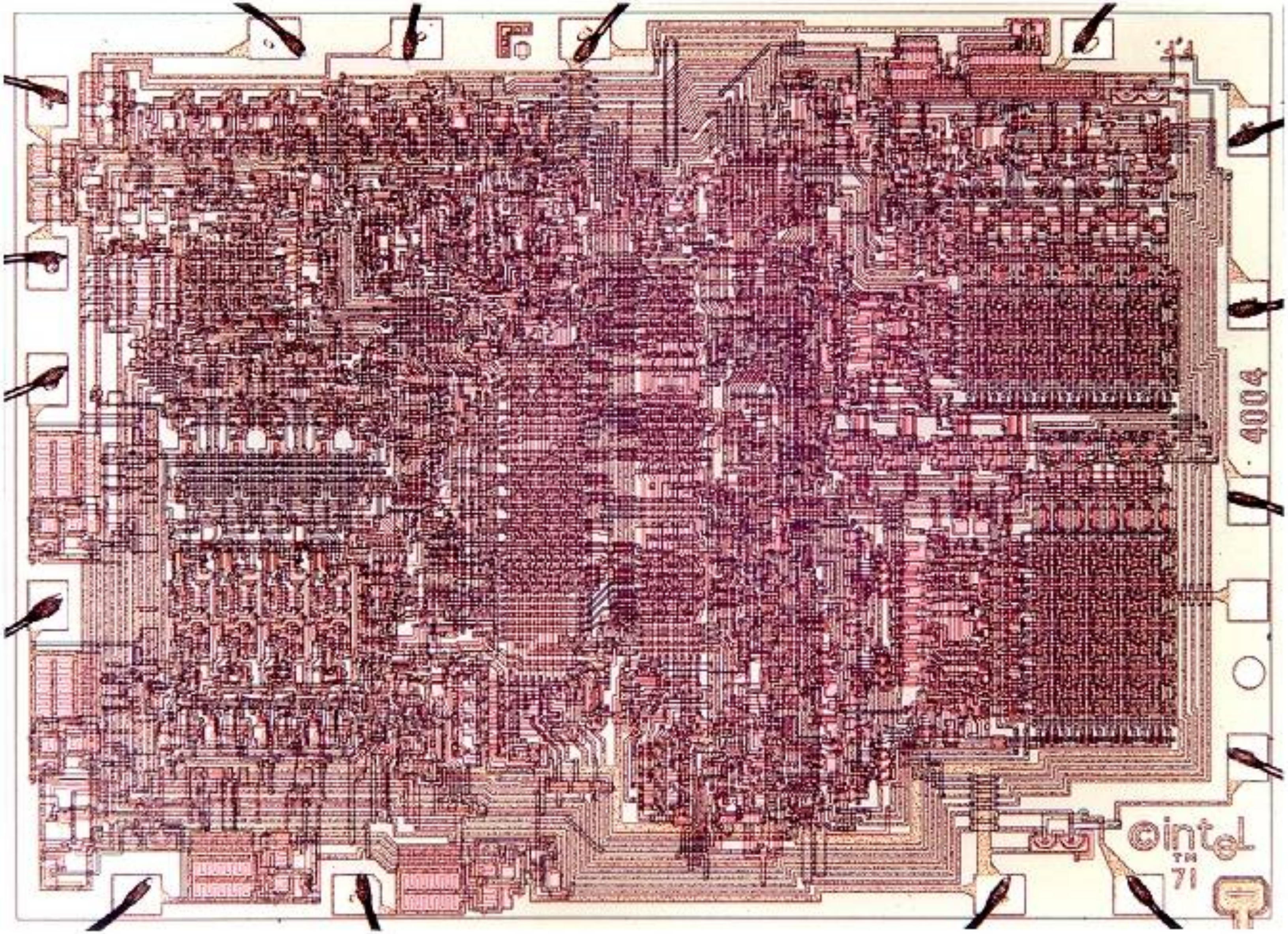
PC ← PC + 1



Processors (Intel 4004, 1971)

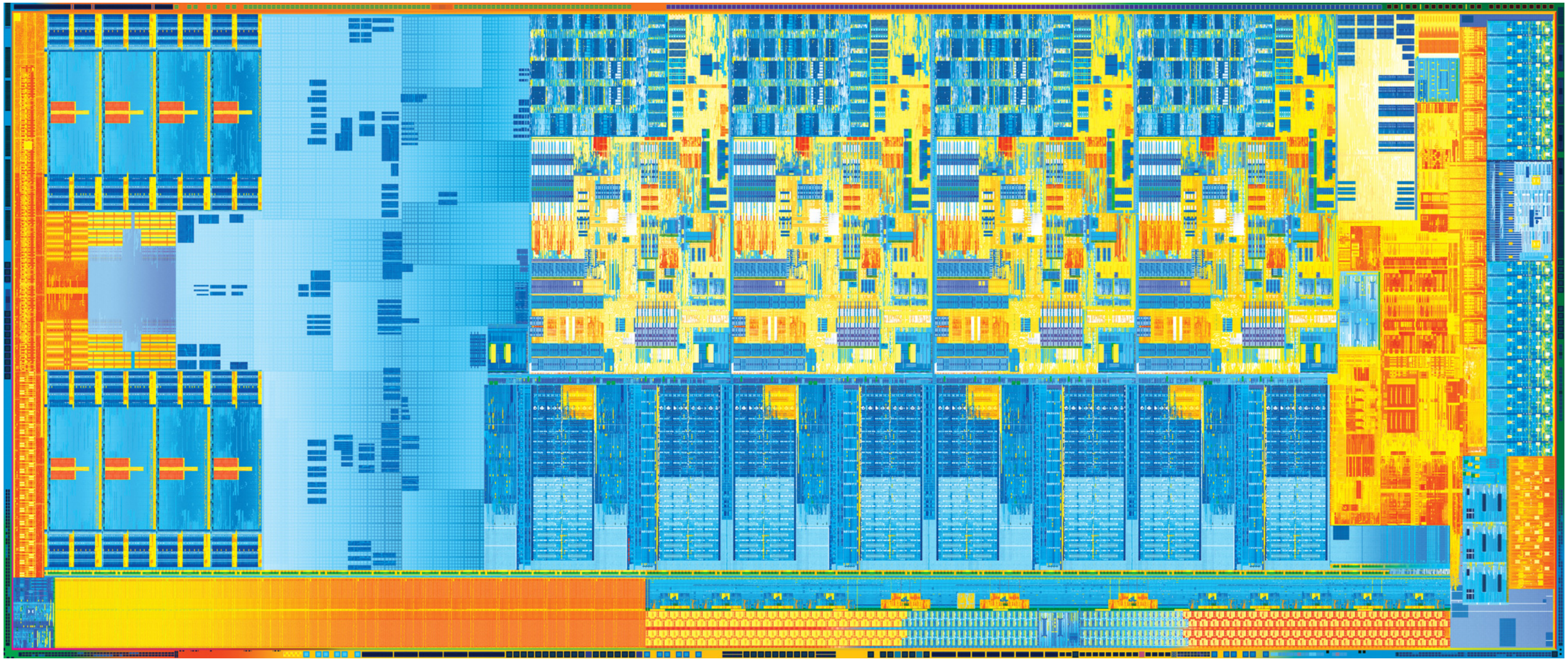


Intel 4004 - 1971



740kHz, 2250 transistors

Intel Ivy Bridge - 2013



3.5GHz, 1.4 Billion transistors

IAS – Instruction Set

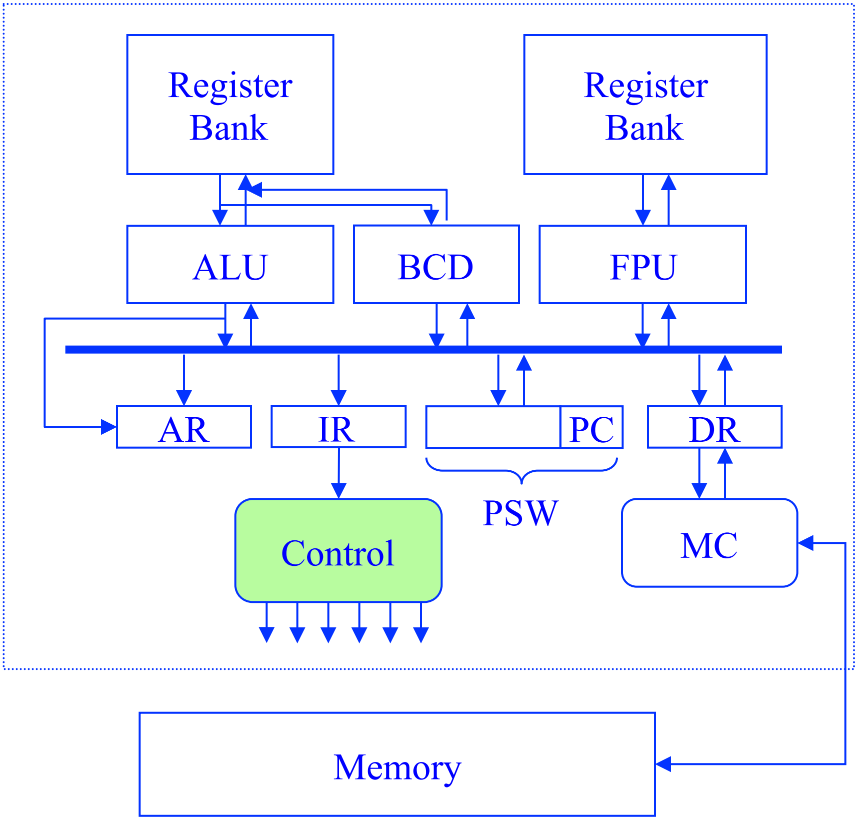
- Very few instructions

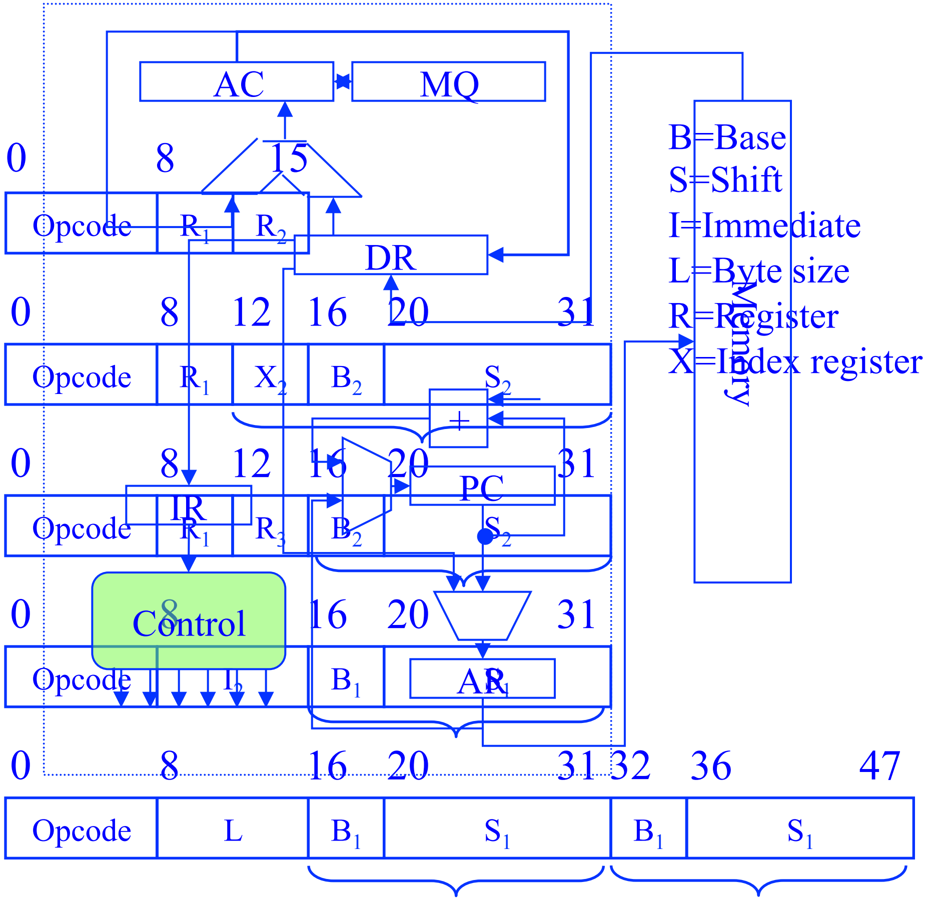
- `AC ← M(X)`
- `M(X) ← AC`
- `goto M(X)`
- `if AC ≥ 0 goto M(X)`
- `AC ← AC +/- M(X)`
- `AC.MQ ← MQ ×/÷ M(X)`
- `AC ← AC <</>> 1`
- `M(X,8:19) ← AC(0:11)`

IBM S/360 – 1964

- Huge number of instructions
  - type of data
  - methods for accessing memory
  - variable-size instructions
  - diversity of operations
  - compound instructions
  - …

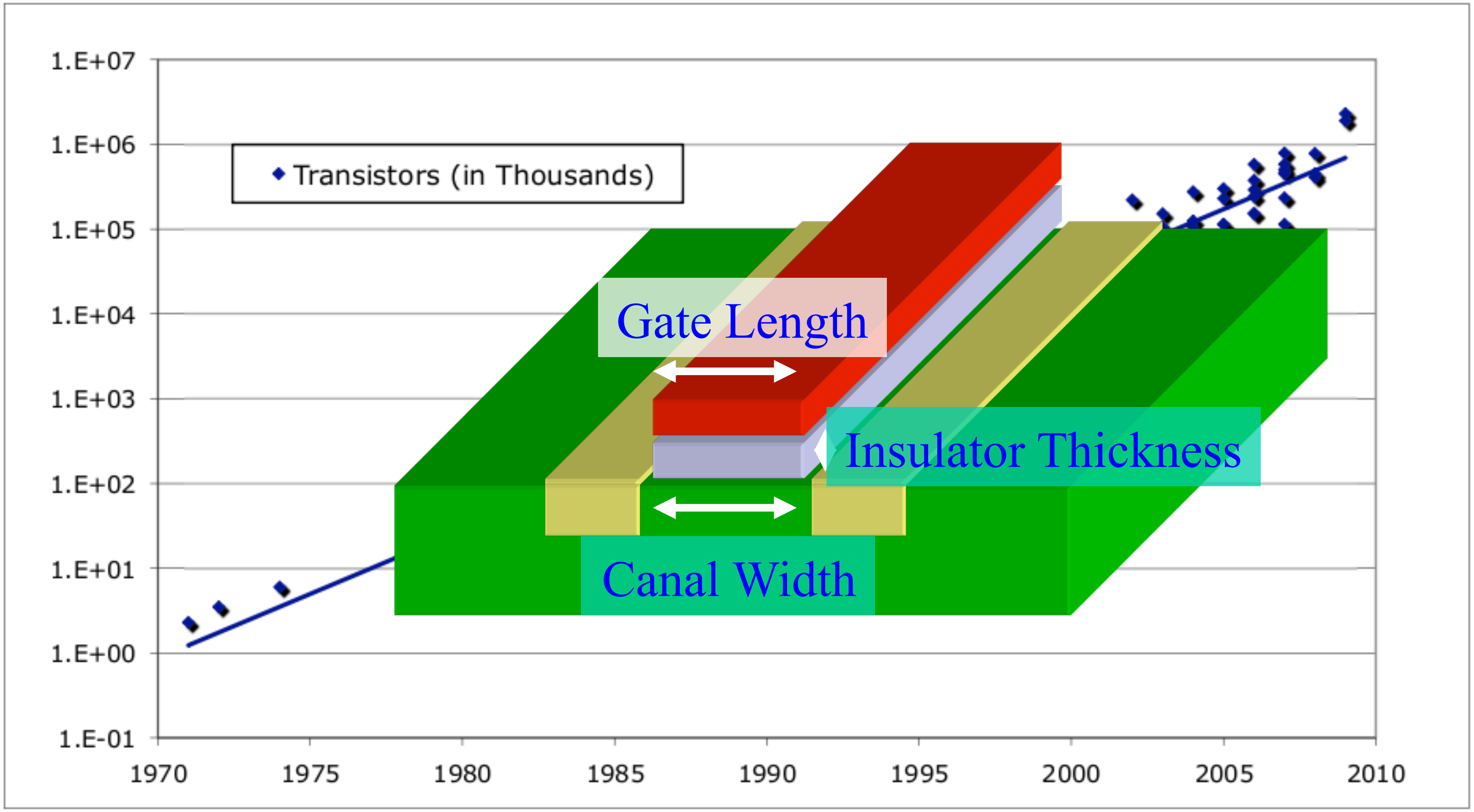
IAS





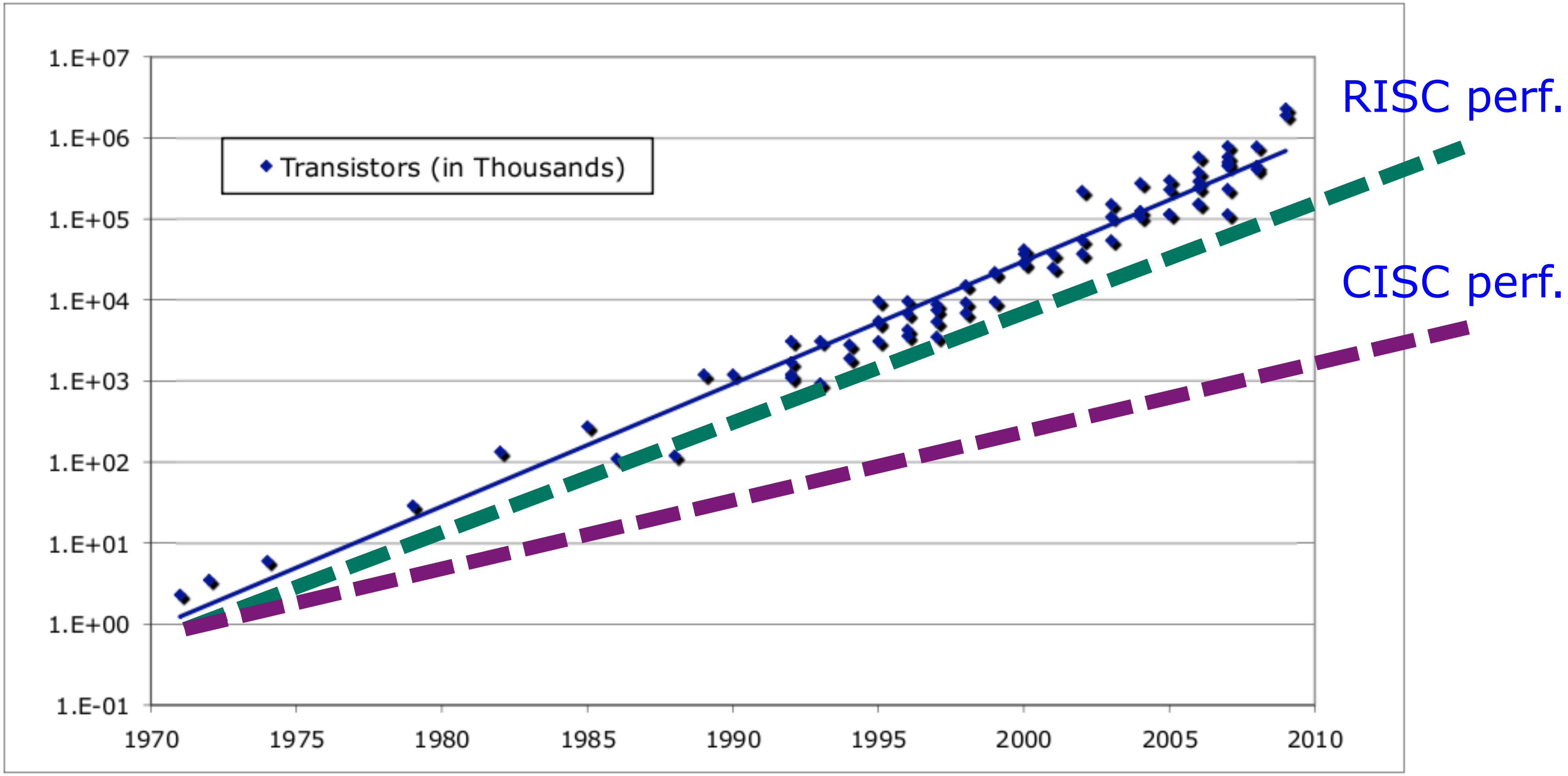
IBM S/360

It’s All About the Technology



- More transistors
- **Faster switching (faster clock)**

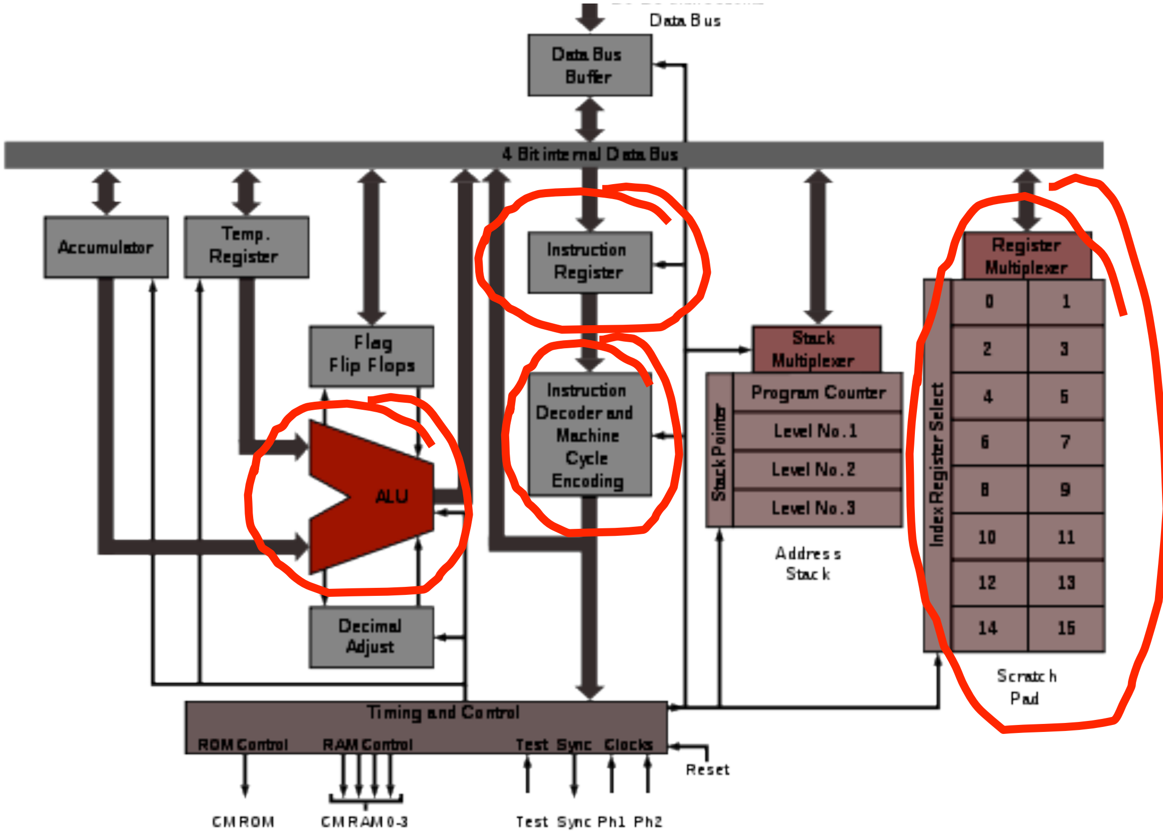
CISC vs. RISC



- CISC: **Complex** Instruction Set Computer
- RISC: **Reduced** Instruction Set Computer

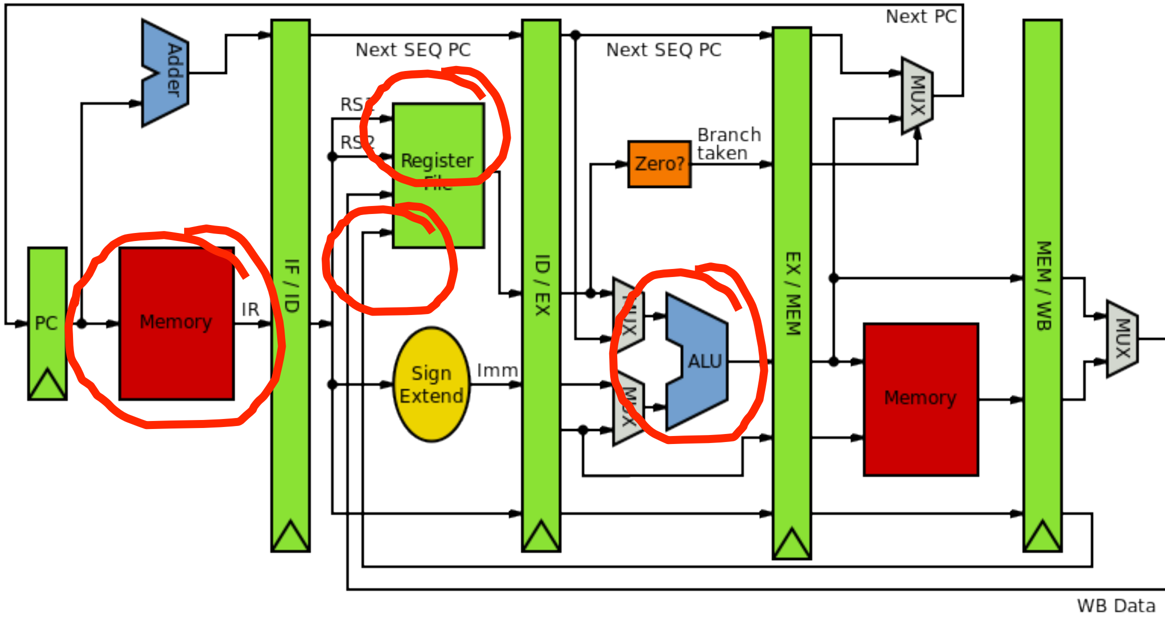
Long Clock Cycle ≈ One Instruction

- Fetch instruction
- Decode instruction
- Compute
- Write result



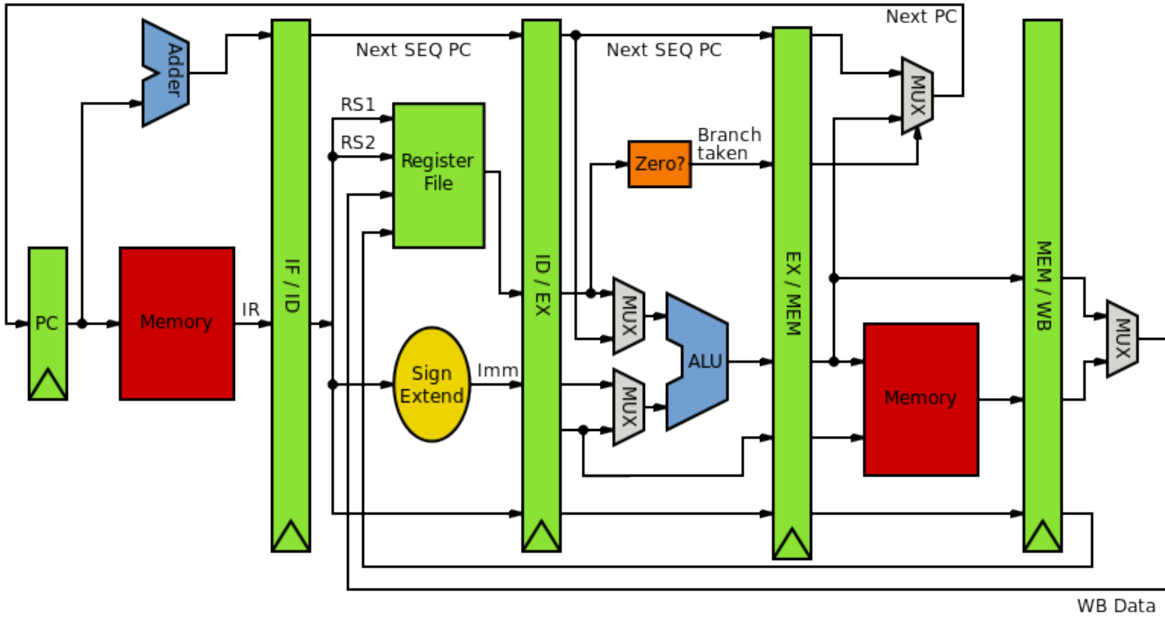
But Speed = Short Cycles

- Fetch instruction
- Decode instruction
- Compute
- Write result



Fast Clock = Pipelining

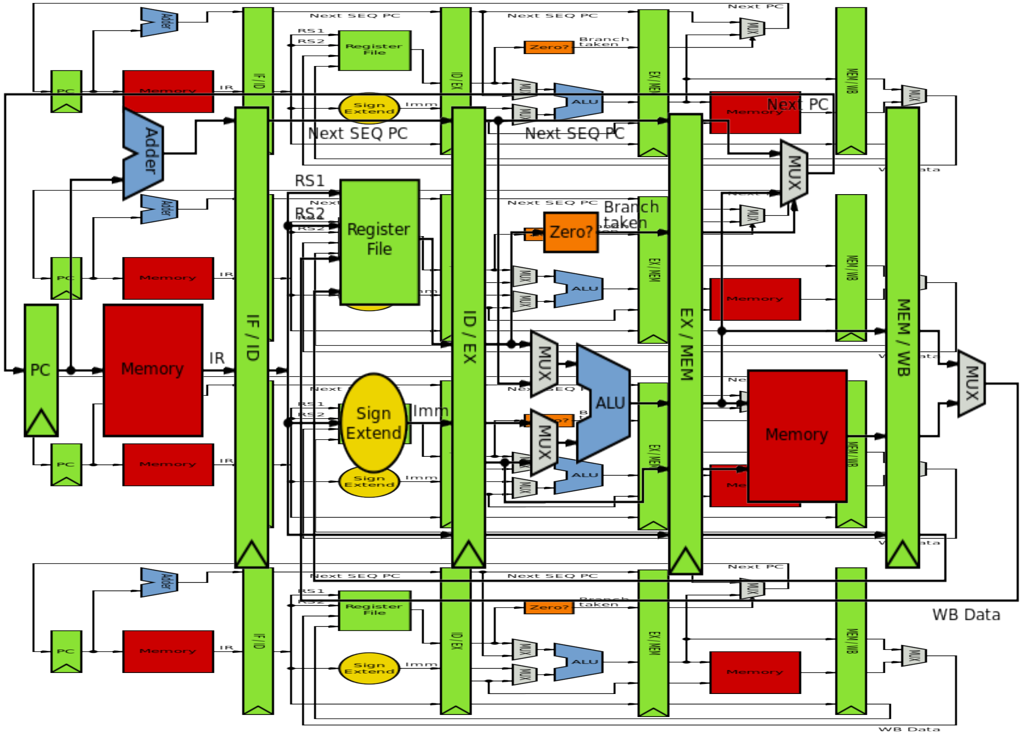
- Fetch instruction
- Decode instruction
- Compute
- Write result



Instruction 1	Inst 1	Inst 1	Inst 1	Inst 1
Instruction 2	Inst 2	Inst 2	Inst 2	
Instruction 3	Inst 3	Inst 3		
Instruction 4	Inst 4			

Architects More (Too ?) Ambitious

- **Superscalar processors**: Several instructions in parallel

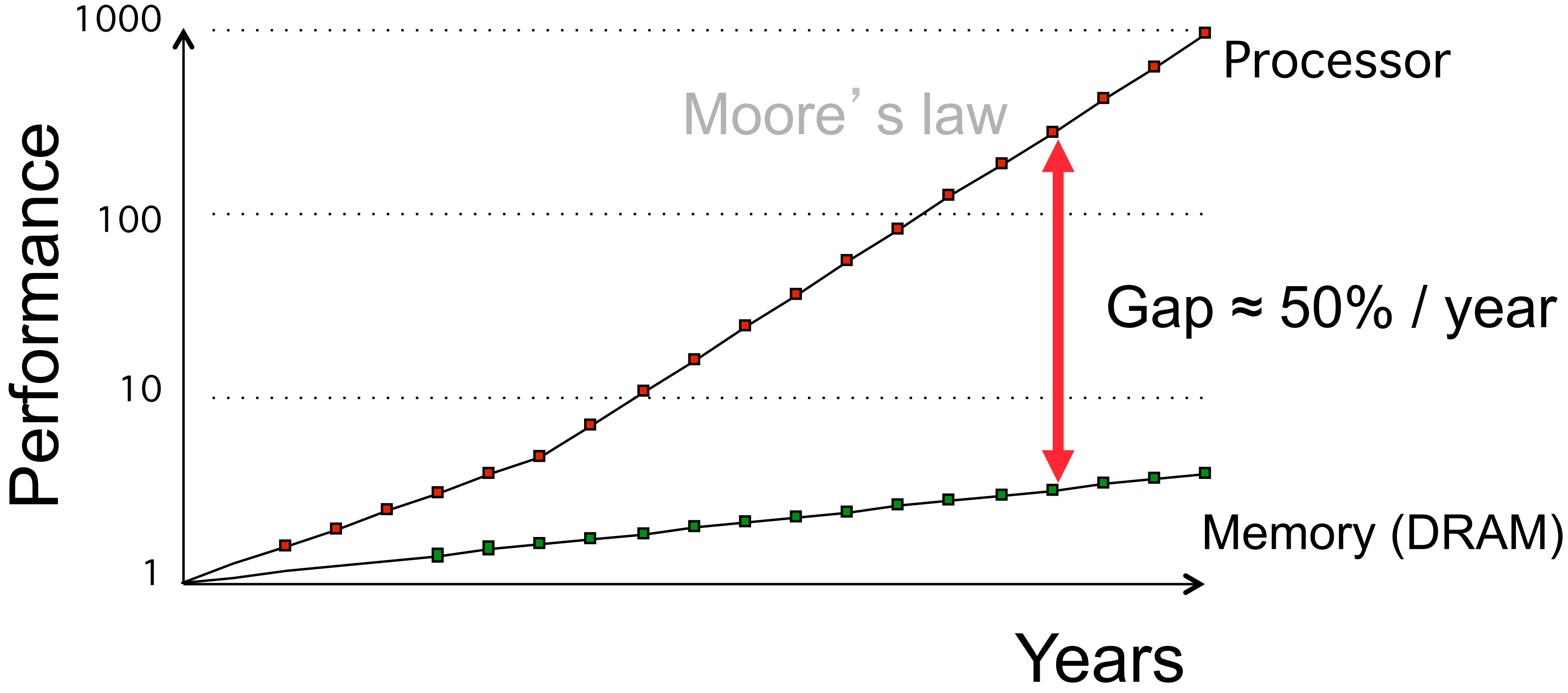


Inst 1
Inst 2
Inst 3
Inst 4

Inst 5
Inst 6
Inst 7
Inst 8

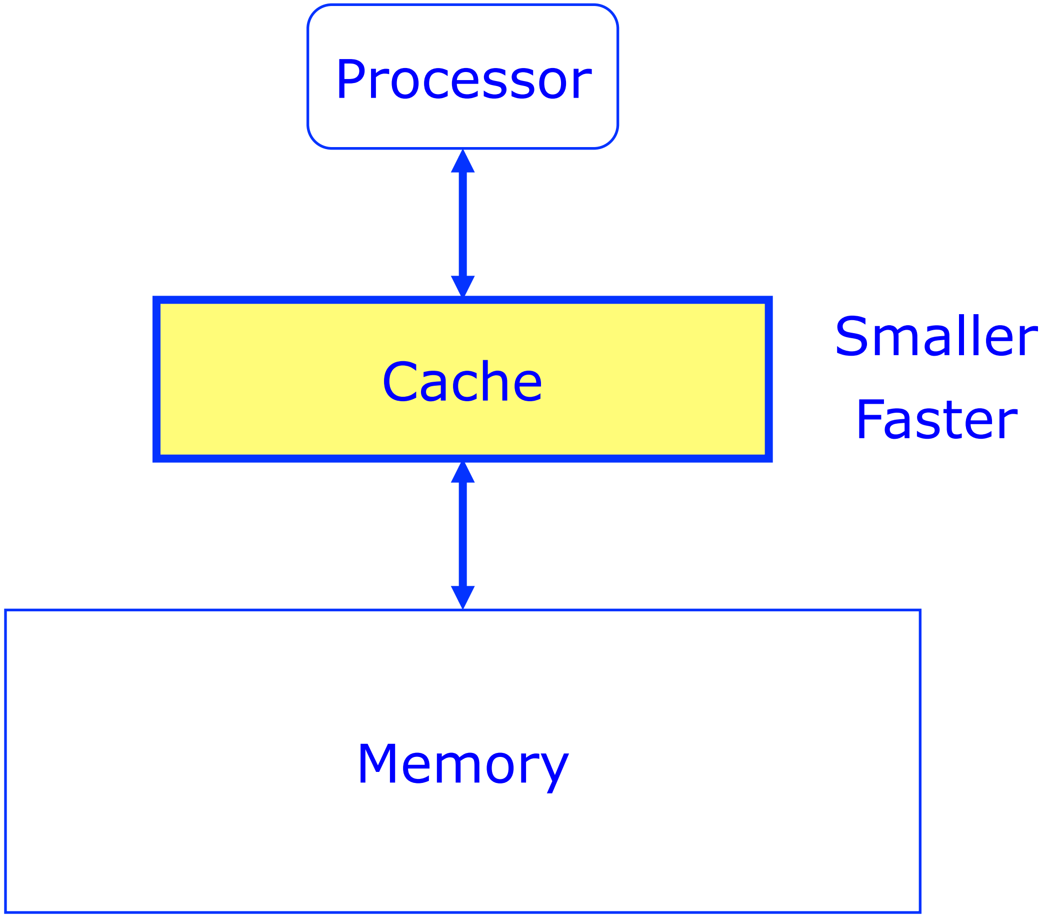
Inst 1
Inst 2
Inst 3
Inst 4

Processor Too Fast vs. Memory



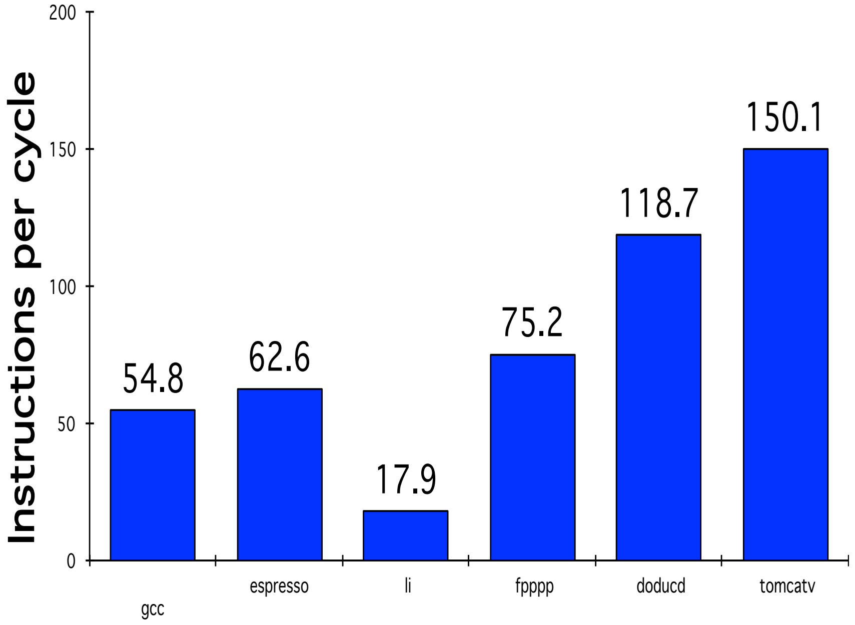
- Increasing Processor/Memory gap

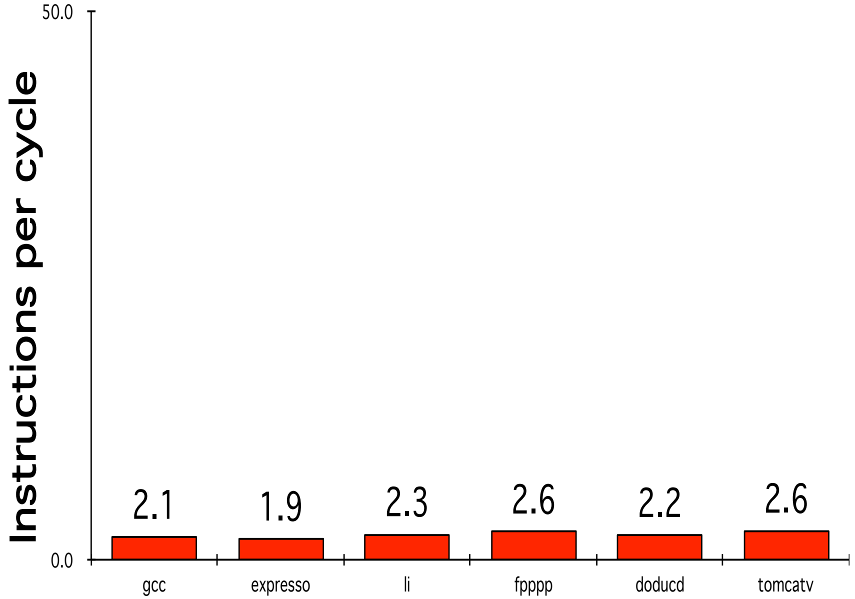
Bridge Gap: Cache Memory



- Instructions and data reused many times

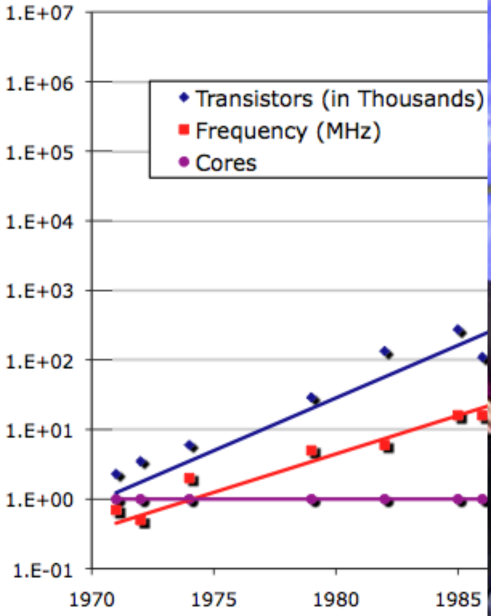
Performance (In Theory…) Performance (…In Practice)

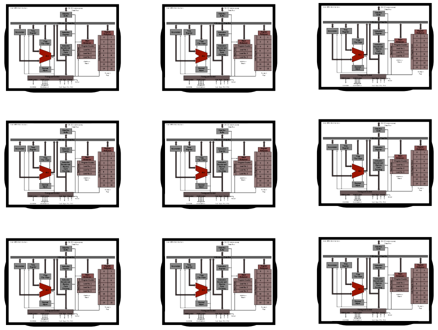




- But, cycle: **1.3x10^{-6} sec. ➡ 2.8x10^{-10} sec.**

1st Technology Shock: Clock Scaling





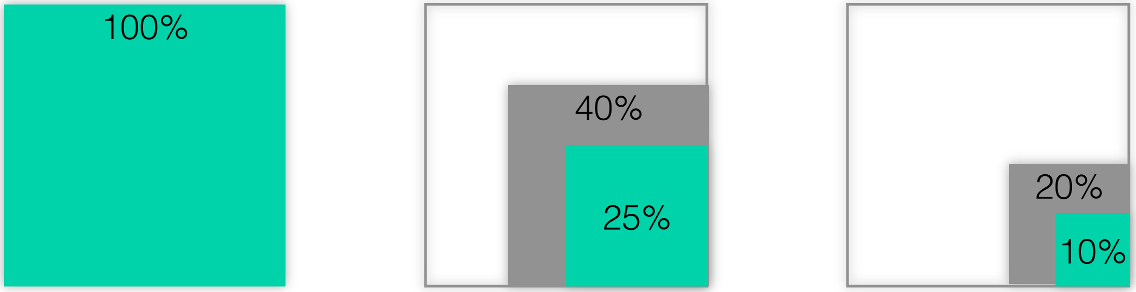
Multi-Cores
GPUs

- Nb transistors keeps increasing

2nd Shock: Dark Silicon

Technology node	45nm	22nm	11nm
#Transistors scaling:		x4	x16
Power scaling:			
@45nm freq		2/3	1/3
@peak freq		1	2/3

Exploitable silicon



Emre Ozer, ARM, Consultation meeting, Brussels, Nov. 2009

- Cannot use all transistors
- Multi-Cores ?

United Business Media

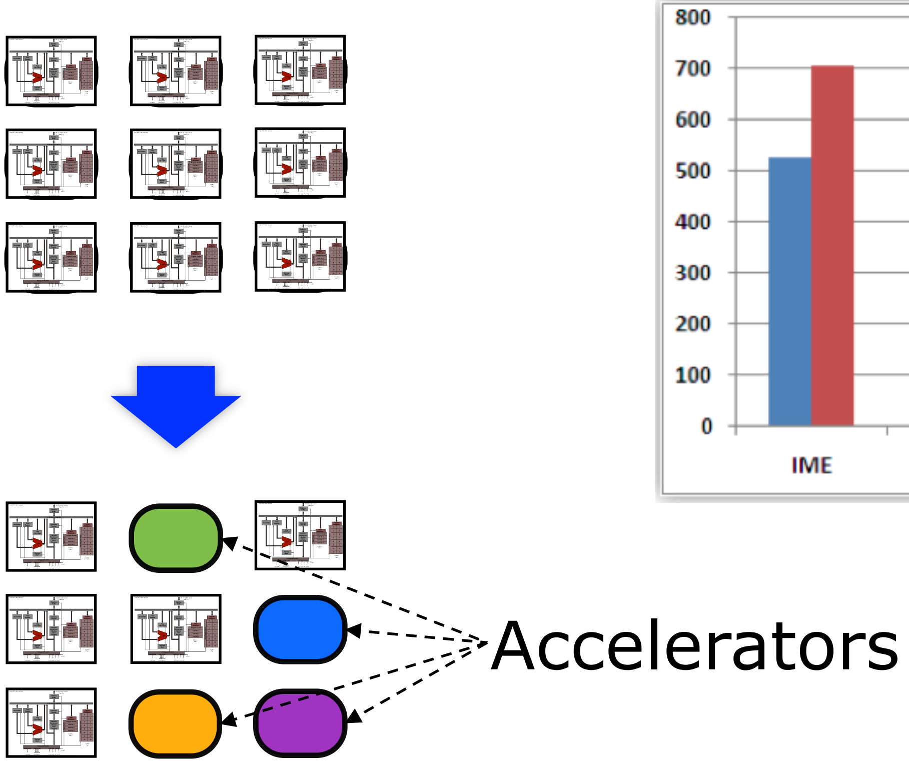
EE Times

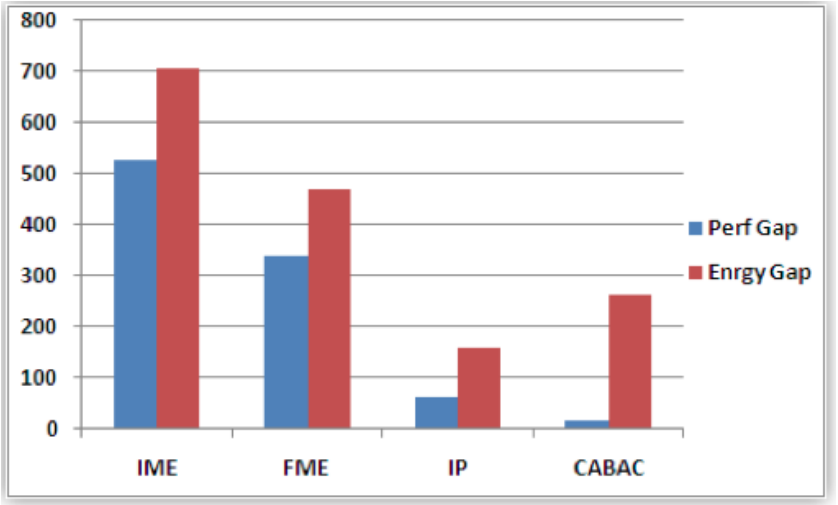
EE Times: Latest News

ARM CTO warns of dark silicon

Keynoter claims it may be impossible to utilize all available transistors on a chip

Specialization: Towards Heterogeneous Multi-Cores

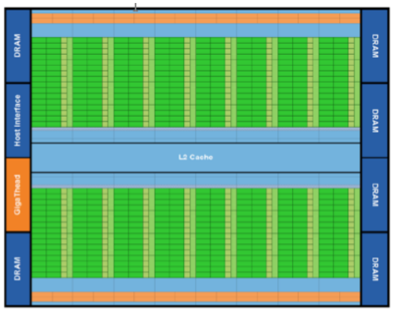




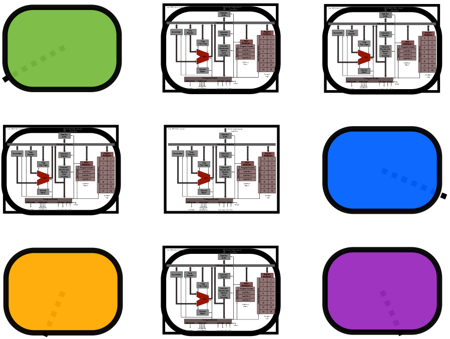
Understanding Sources of Inefficiency in General-Purpose Chips, Rehan Hameed et al., ISCA 2010

- Huge energy ratio proc./specialized circ.
- Compatible with Dark Silicon

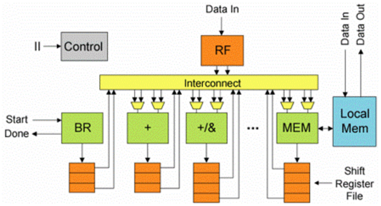
Which Accelerators ?



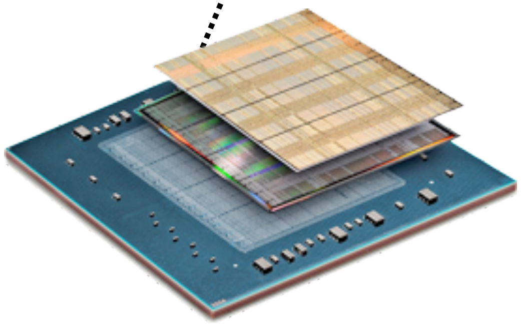
GPUs



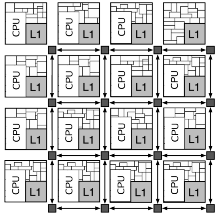
Heterogeneous multi-cores



More efficient
More specialized



FPGAs
CGRAs



- Flexibility/Efficiency tradeoff ?

Thank You !