# Restoring / Nonrestoring division.

Q : Dividend.

M : Divisor.

n : Cycles } # bits in dividend.

## RECAP

## Contemporary Microprocessor Design

- Instruction Set Processors [ISPs]
  - ISA
    - Specification.

① 4-bit

8-bit

↳ PC

② 32-bit

- Instruction Set
- ISA
- Instruction Pipelining
- Cache memory
- Workstations

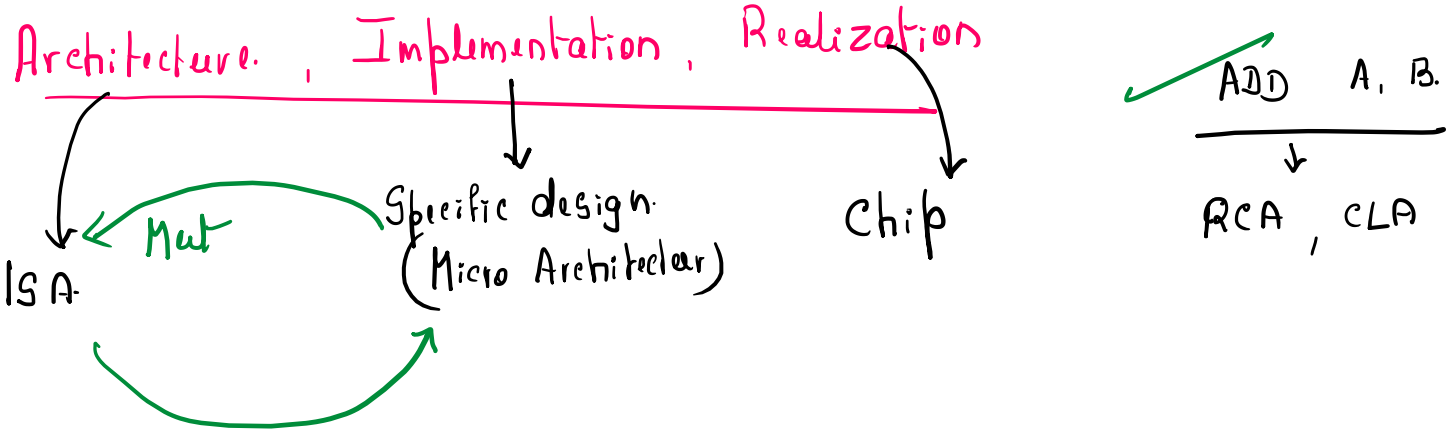
③ → Fast-clock

→ Deeply pipelined.

→ Out-of-execution. } Pipeline stalls
→ Branch prediction

④ MPs

ILP (Parallelism) — thread lvl parallelism (TLP)
ILP (Parallelism) — Memory level (MLP)



## High Performance

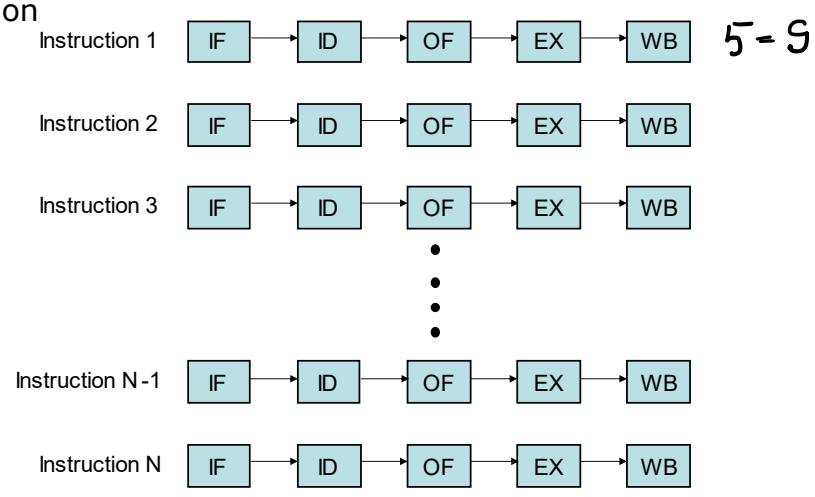
$$T = \frac{N \times S}{R}$$

- Reduce the value of *T*
  - Reduce *N* and *S*
  - Increase *R*
- Reduction in *N* – complexity of instruction increases
  - *S* increases
- Increasing *R*-using higher frequency clock
  - Time required to complete a basic execution step reduces
- *N,S*, and *R* are not independent parameters
  - Changing one may affect the other
- **Attempt to improve performance only by overall reduction of *T***



## Further Performance Improvement?

- Instructions are executed one after another
- *S* is the total number of basic steps (clock cycles) required to execute an instruction




5 - S

ADD [R1], [R2]

ADD R1, R2

N × 5

# Overlapping the Execution of Successive Instructions

Clock cycle →

Instruction ↓

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF | ID | OF | EX | WB | | | | | | | | | |
| I2 | | IF | ID | OF | EX | WB | | | | | | | | |
| I3 | | | IF | ID | OF | EX | WB | | | | | | | |
| I4 | | | | IF | ID | OF | EX | WB | | | | | | |
| I5 | | | | | IF | ID | OF | EX | WB | | | | | |
| I6 | | | | | | IF | ID | OF | EX | WB | | | | |
| I7 | | | | | | | IF | ID | OF | EX | WB | | | |
| I8 | | | | | | | | IF | ID | OF | EX | WB | | |
| I9 | | | | | | | | | IF | ID | OF | EX | WB | |
| I10 | | | | | | | | | | IF | ID | OF | EX | WB |

- One instruction completed in every clock cycle from 5th clock cycle onwards
- **For the purpose of computing *T*, the effective value of *S* is 1**



# Instruction Pipelining

Clock cycle →

Instruction ↓

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF | ID | OF | EX | WB | | | | | | | | | |
| I2 | | IF | ID | OF | EX | WB | | | | | | | | |
| I3 | | | IF | ID | OF | EX | WB | | | | | | | |
| I4 | | | | IF | ID | OF | EX | WB | | | | | | |
| I5 | | | | | IF | ID | OF | EX | WB | | | | | |
| I6 | | | | | | IF | ID | OF | EX | WB | | | | |
| I7 | | | | | | | IF | ID | OF | EX | WB | | | |
| I8 | | | | | | | | IF | ID | OF | EX | WB | | |
| I9 | | | | | | | | | IF | ID | OF | EX | WB | |
| I10 | | | | | | | | | | IF | ID | OF | EX | WB |

- One instruction completed in every clock cycle from 5th clock cycle onwards
- **For the purpose of computing *T*, the effective value of *S* is 1**

## Superscalar Execution

- Multiple instruction pipelines
  - Multiple functional units
- Execution of several instructions per clock cycle
- Effective value of *S* can be reduced to **less than one**
- **Many of today's high performance processors are designed in this manner**



→ K-fold improvement

K: No of stages

K = 5

Pipeline Idealism

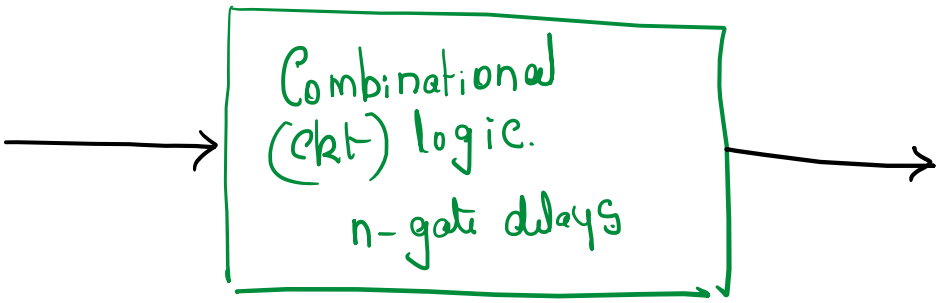
- Uniform Subcomputation
- Identical Computation.
- Independent Computation.

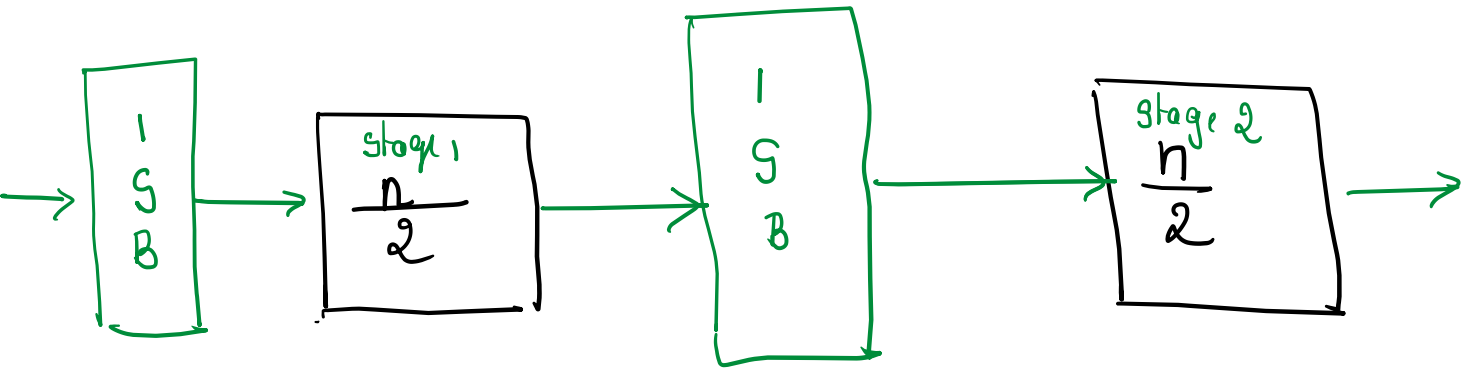
Pipeline.

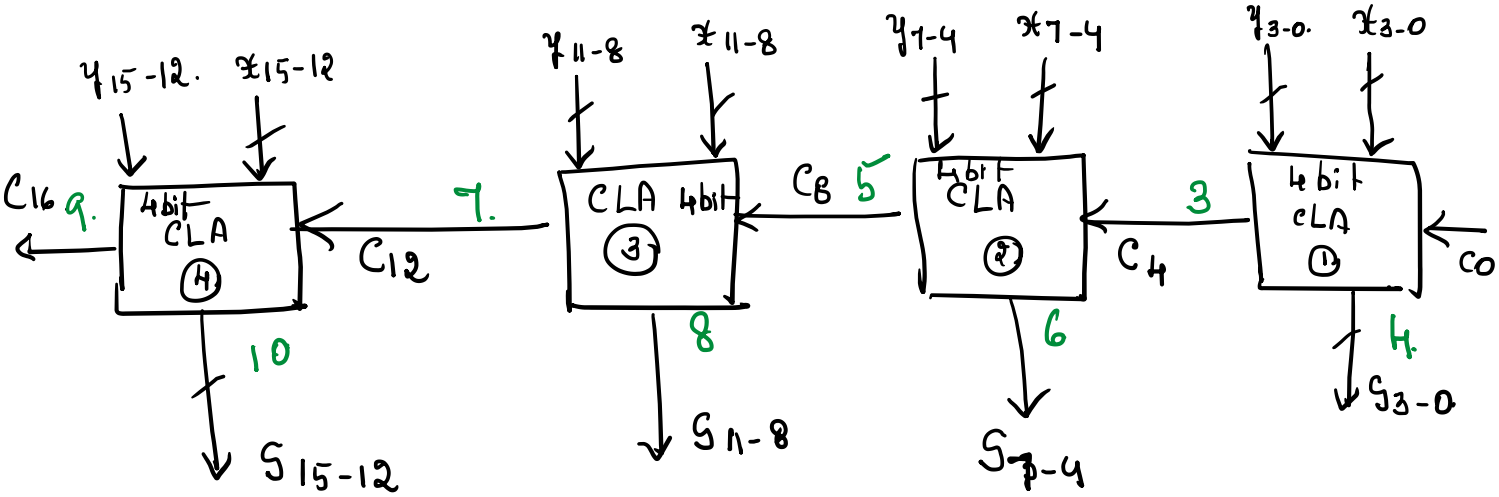
Arithmetic.

Instruction.

# Pipelined Adder




X + Y

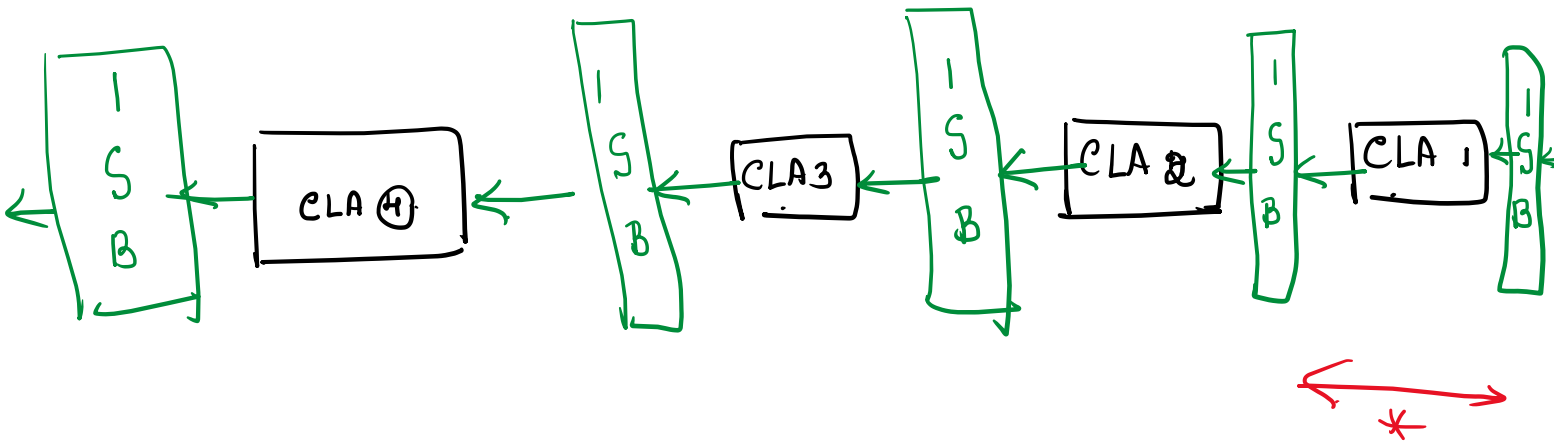
M = 4

$A_1$ X Y

$A_2$ A B

$A_3$ C D

$A_4$ E F




Delay/Latency in a stage = Logic delay of ckt in stage + Delay in ISB

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $A_1(X_1, Y)$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ | | | |
| $A_2(A_1, B)$ | | $S_{21}$ | $S_{22}$ | $S_{23}$ | $S_{24}$ | | |
| $A_3(C_1, D)$ | | | $S_{31}$ | $S_{32}$ | $S_{33}$ | $S_{34}$ | |
| $A_4(E_1, F)$ | | | | $S_{41}$ | $S_{42}$ | $S_{43}$ | $S_{44}$ |

$\rightarrow$ $4 \times 4 \approx 16$.

$K = 4$.

M-additions = $(M + K - 1)$

Total: $(M + K - 1)$ * Latency of a stage.

Latency of non Pipelined adder: $T$

M computation $= \underline{M \times T}$

Pipeline stage $= \frac{T}{K} + Rd.$

Pipelined ckt latency $= (M+K-1) * \left(\frac{T}{K} + Rd\right)$

$M >> K.$

$\frac{T}{K} >> Rd.$

$\approx \underline{\frac{MT}{K}}$ $(M \times T)$

## INSTRUCTION PIPELINING

- Instruction cycle.
  - Inst Fetch - IF
  - Inst Decode - ID
  - Operand fetch - OF|RD
  - Perform Operation - EX
  - Store the result - WB

} Logical steps / Basic

$I_1$ $I_2$ . . . $I_8$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF1 | ID1 | RD1 | EX1 | WB1 | | | | | | | | | | |
| I2 | | IF2 | ID2 | RD2 | EX2 | WB2 | | | | | | | | | |
| I3 | | | IF3 | ID3 | RD3 | EX3 | WB3 | | | | | | | | |
| I4 | | | | IF4 | ID4 | RD4 | EX4 | WB4 | | | | | | | |
| * | | | | IF4 | IF4 | IF4 | ID4 | RD4 | EX4 | WB4 | | | | | |
| I5 | | | | | IF5 | ID5 | RD5 | EX5 | WB5 | | | | | | |
| | | | | | • | • | IF5 | ID5 | RD5 | EX5 | WB5 | | | | |
| I6 | | | | | | IF6 | ID6 | RD6 | EX6 | WB6 | | | | | |
| | | | | | | • | • | IF6 | ID6 | RD6 | EX6 | WB6 | EX6 | EX6 | WB6 |
| I7 | | | | | | | IF7 | ID7 | RD7 | EX7 | WB7 | | | | |
| | | | | | | | • | • | IF7 | ID7 | RD7 | • | • | • | EX7 |
| I8 | | | | | | | | IF8 | ID8 | RD8 | EX8 | WB8 | | | |
| | | | | | | | | | | | | | | | |

Memory fetch.

Multiplication

SUB

Filling — Full. — Drained.

$N+(K-1)$

$8+(5-1) = 12$

$M+(K-1)$

→ Multicycle operation. | More than one cycle.
→ Memory operation.

EX: Ideal

| | | | | | | EX |
|---|---|---|---|---|---|---|
| I1 | ADD | R2 | R1 | R0 | 1 | 1 |
| I2 | MUL | R5 | R3 | R4 | 1 | 3 |
| I3 | FMUL | F0 | F1 | F2 | 1 | 5 |
| I4 | FADD | F5 | F3 | F4 | 1 | 4 |
| I5 | LOAD | R6, | X. | | 1 | 1 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF1 | ID1 | RD1 | EX1 | WB1 | | | | | | | | | | |
| I2 | | IF2 | ID2 | RD2 | EX2 | WB2 | | | | | | | | | |
| | | IF2 | ID2 | RD2 | EX2 | EX2 | EX2 | WB2 | | | | | | | |
| I3 | | | IF3 | ID3 | RD3 | EX3 | WB3 | | | | | | | | |
| | | | IF3 | ID3 | RD3 | - | - | EX3 | EX3 | EX3 | EX3 | EX3 | WB3 | | |
| I4 | | | | IF4 | ID4 | RD4 | EX4 | WB4 | | | | | | | |
| | | | | IF4 | ID4 | - | - | RD4 | - | - | - | - | EX4 | EX4 | EX4 |
| I5 | | | | | IF5 | ID5 | RD5 | EX5 | WB5 | | | | | | |
| | | | | | IF5 | - | - | ID5 | - | - | - | - | RD5 | - | - |
| | | | | | | | | | | | | | | | |

# of Clock cycles = 18