CS222: Processor Design

Dr. A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati

Outline

- Previous Class:
  - Floating points: Add, Sub, Mul,Div, Rounding
- Processor Design: Introduction
  - building blocks
- A simple implementation: Single Cycle
  - Data path and control
- Performance considerations
- Multi-cycle design
  - Data path and control
- Micro-programmed control
- Exception handling

FP numbers with base = 2

$$(-1)^S \times F \times 2^E$$

S = Sign

F = Fraction (fixed point number)

usually called **Mantissa** or S**ignificand**

E = Exponent (positive or negative integer)

- How to divide a word into S, F and E?
- How to represent S, F and E?

Value Range for F

- Single precision numbers

$$1 \leq F \leq 2 - 2^{-23} \quad \text{or} \quad 1 \leq F < 2$$

- Double precision numbers

$$1 \leq F \leq 2 - 2^{-52} \quad \text{or} \quad 1 \leq F < 2$$

These are “normalized”.

Value Range for E

- Single precision numbers

  $-126 \leq E \leq 127$

  (all 0's and all 1's have special meanings)

- Double precision numbers

  $-1022 \leq E \leq 1023$

  (all 0's and all 1's have special meanings)

Overflow and underflow

largest positive/negative number (SP) =

$\pm(2 - 2^{-23}) \times 2^{127} \cong \pm 2 \times 10^{38}$

smallest positive/negative number (SP) =

$\pm 1 \times 2^{-126} \cong \pm 2 \times 10^{-38}$

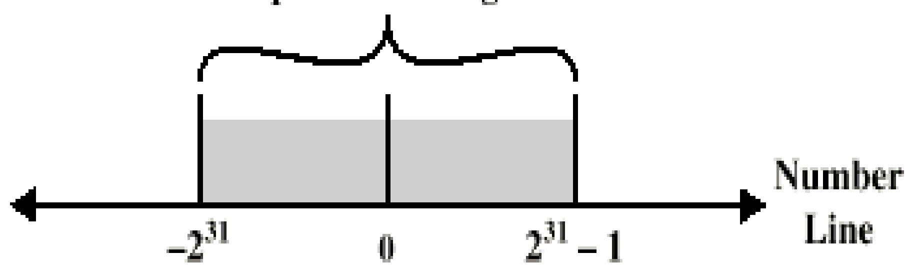
Largest positive/negative number (DP) =

$\pm(2 - 2^{-52}) \times 2^{1023} \cong \pm 2 \times 10^{308}$

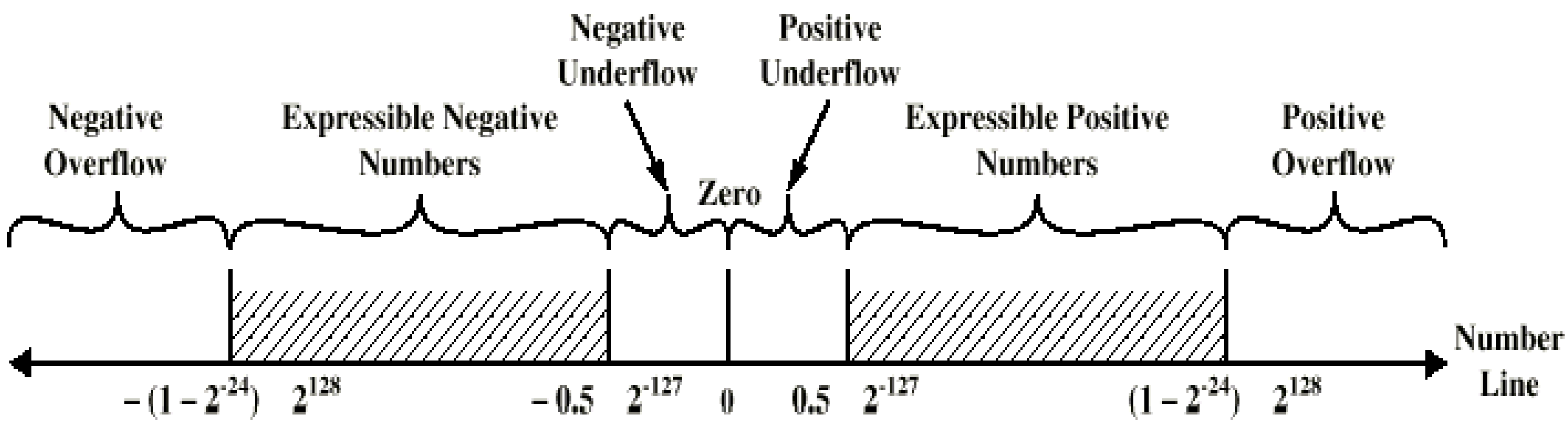
Smallest positive/negative number (DP) =

$\pm 1 \times 2^{-1022} \cong \pm 2 \times 10^{-308}$

Expressible Numbers



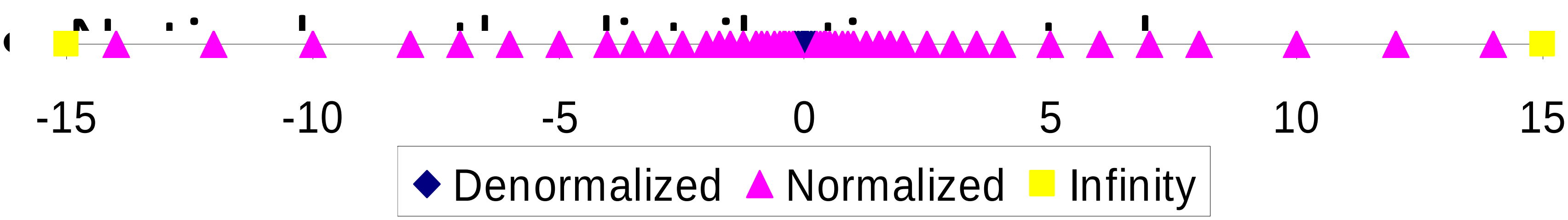
(a) Twos Complement Integers



(b) Floating-Point Numbers

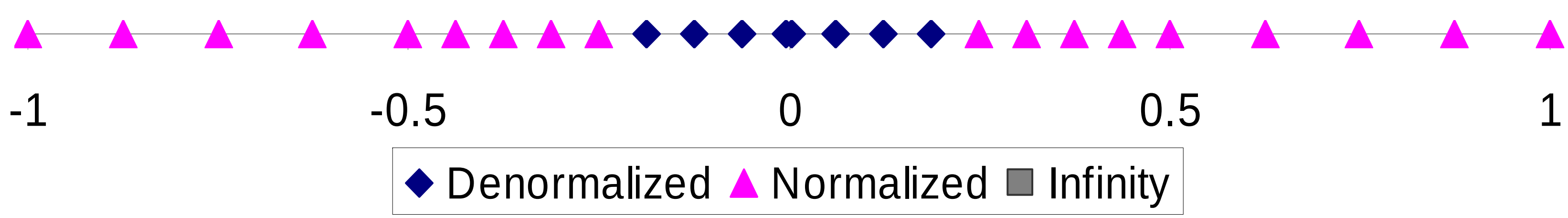
Distribution of Values

- 6-bit IEEE-like format
  - e = 3 exponent bits
  - f = 2 fraction bits
  - Bias is 3



Distribution of Values (close-up view)

- 6-bit IEEE-like format
  - e = 3 exponent bits
  - f = 2 fraction bits
  - Bias is 3



Floating point operations: ADD

- Add/subtract $A = A1 \pm A2$

$[(-1)^{S1} \times F1 \times 2^{E1}] \pm [(-1)^{S2} \times F2 \times 2^{E2}]$

suppose $E1 > E2$, then we can write it as

$[(-1)^{S1} \times F1 \times 2^{E1}] \pm [(-1)^{S2} \times F2' \times 2^{E1}]$

where $F2' = F2 / 2^{E1-E2}$,

The result is

$(-1)^{S1} \times (F1 \pm F2') \times 2^{E1}$

It may need to be normalized

Floating point operations

- Multiply

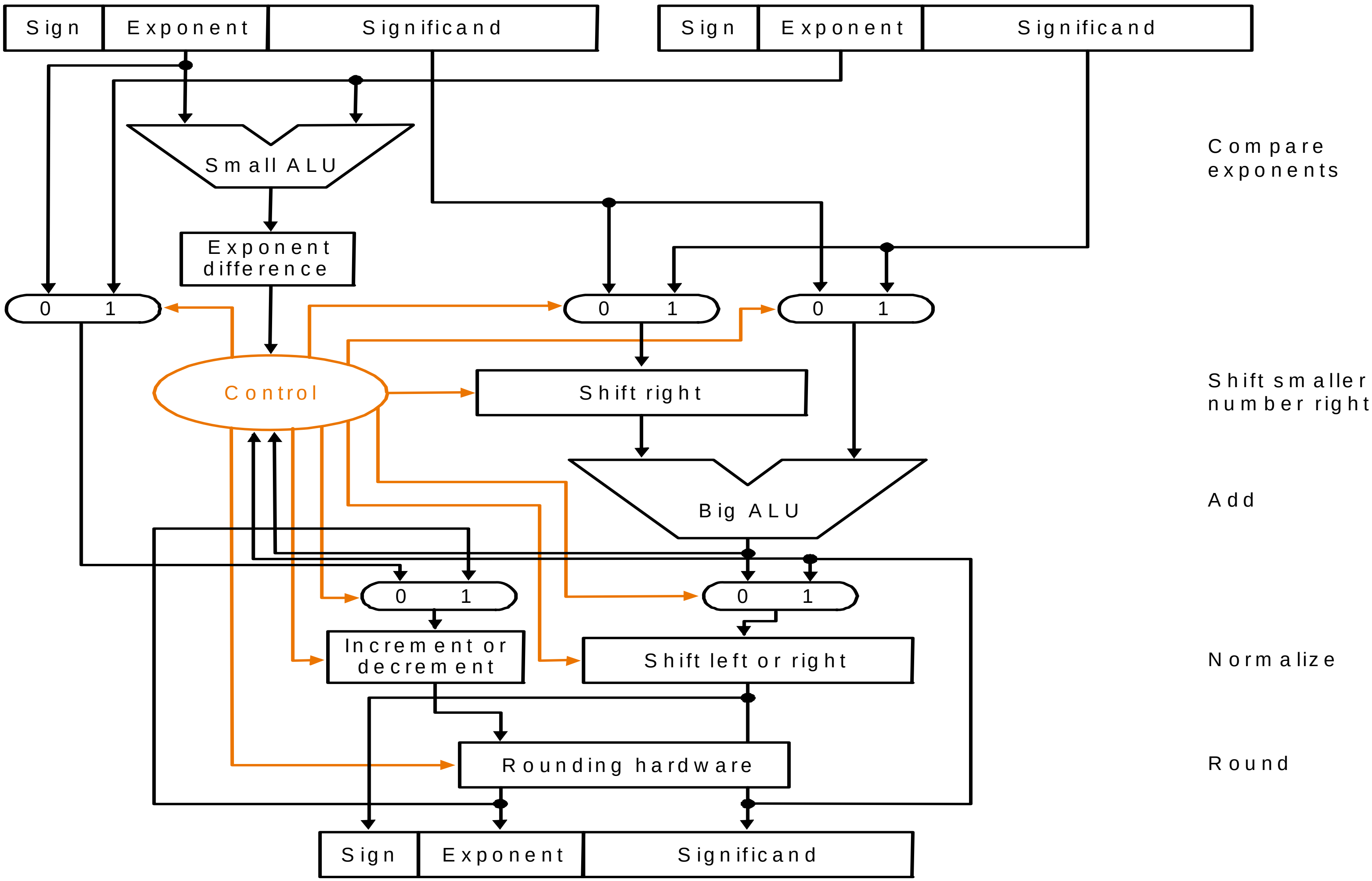
$[(-1)^{S1} \times F1 \times 2^{E1}]$ **x** $[(-1)^{S2} \times F2 \times 2^{E2}]$

$= (-1)^{S1 \oplus S2} \times (\mathbf{F1xF2}) \times 2^{E1+E2}$

Since $1 \leq (F1xF2) < 4$,

the result may need to be normalized

Float Multiplication



Floating point operations

- Divide

$[(-1)^{S1} \times F1 \times 2^{E1}] \div [(-1)^{S2} \times F2 \times 2^{E2}]$

$= (-1)^{S1 \oplus S2} \times (\mathbf{F1 \div F2}) \times 2^{E1-E2}$

Since $.5 < (F1 \div F2) < 2$,

the result may need to be normalized

(assume $F2 \neq 0$)

Testing Associatively with FP

- $X= -1.5x10^{38}$, $Y=1.5x10^{38}$, $z=1.0$
- $X+(Y+Z) = -1.5x10^{38} + (1.5x10^{38} + 1.0)$

  $= -1.5x10^{38} + 1.5x10^{38}$

  $=0$
- $(X+Y)+Z = (-1.5x10^{38} + 1.5x10^{38}) + 1.0$

  $= 0.0 + 1.0$

  $=1$

Accuracy

- Precision is lost when some bits are shifted to right of the rightmost bit or are thrown
- Three extra bits are used internally -

  G (guard), R (round) and S (sticky)
  - G and R are simply the next two bits after LSB
  - S = 1 iff any bit to right of R is non-zero

1.	11010110101100010110110	GRS

Rounding using G, R and S

- if G=1 & R=1, add 1 to LSB
- if G=0 & R=0 or 1, no change
- if G=1 & R=0, look at S
  - if S=1, add 1 to LSB
  - if S=0, round to the nearest “even”

    i.e., add 1 to LSB if LSB = 1

FP instructions in MIPS

- 32 floating point registers $f0 . . $f31
- Single precision arithmetic
  - add.s, sub.s, mul.s, div.s, abs.s, neg.s
- Double precision arithmetic (similar)
- Conditional branch
  - bc1t, bc1f, c.lt.s, c.lt.d   (lt|le|gt|ge|eq|ne)
- Conversion
  - cvt.d.s, cvt.d.w, cvt.s.d, cvt.s.w, cvt.w.d, cvt.w.s

Processor Design

- Introduction
  - building blocks
- A simple implementation
  - data path and control
- Performance considerations
- Multi-cycle design
  - data path and control
- Micro-programmed control
- Exception handling

Simple Processor Design

- MIPS subset for implementation
- Design overview
- Division into data path and control
- Building blocks - combinational and sequential
- Clock and timings
- Components required for MIPS subset

MIPS subset for implementation

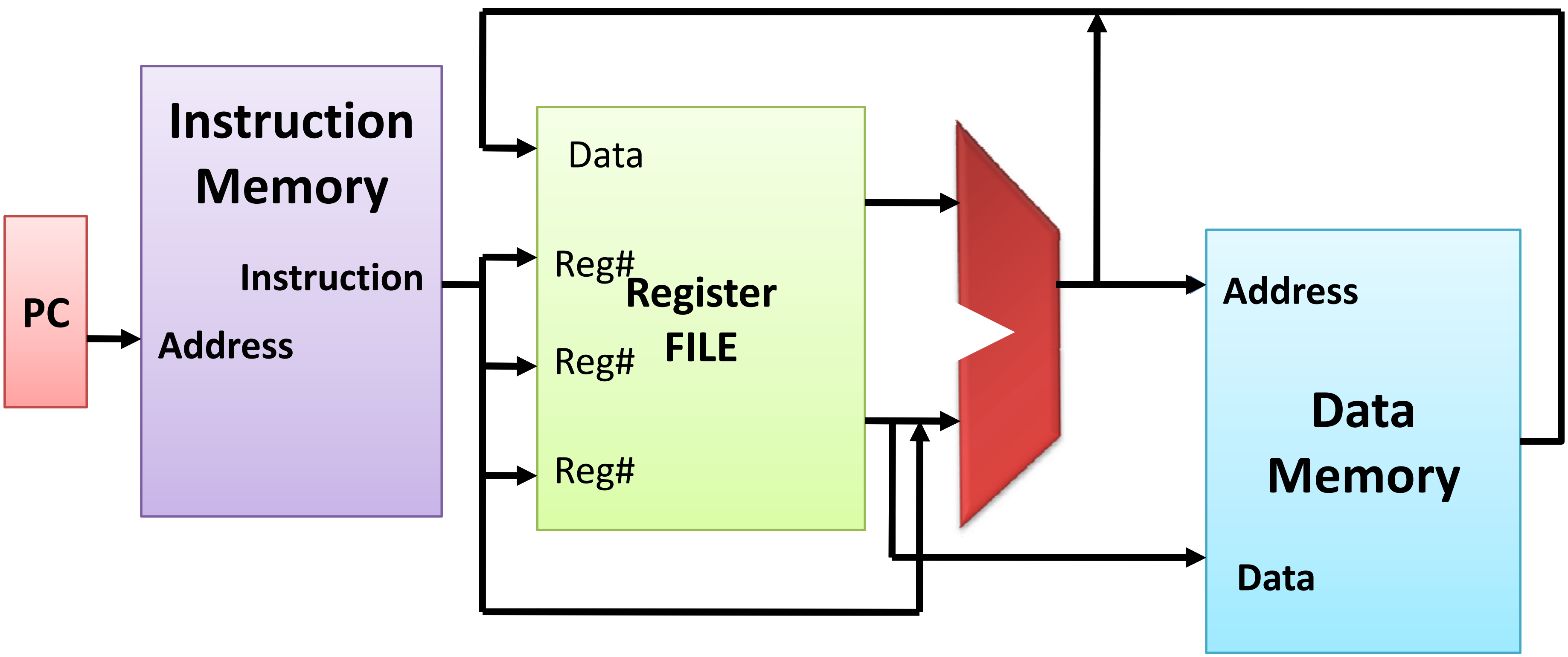
- Arithmetic - logic instructions
  - **add, sub, and, or, slt**
- Memory reference instructions
  - **lw, sw**
- Control flow instructions
  - **beq, j**

Incremental changes in the design to include other instructions will be discussed later

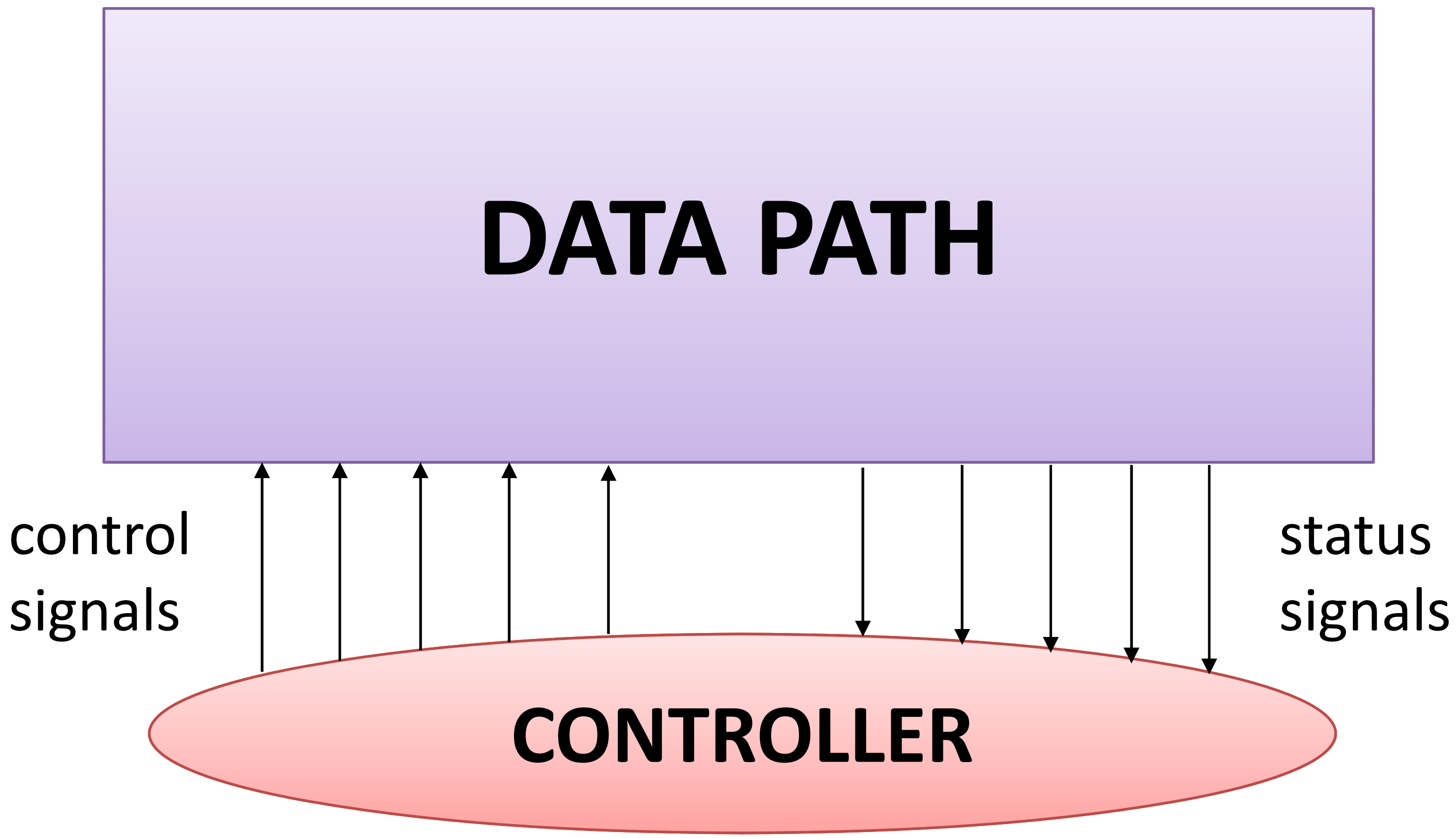
Generic Implementation

- Use the program counter (PC) to supply instruction address
- Get the instruction from memory
- Read registers
- Use the instruction to decide exactly what to do

Design overview



Division into Data path and Control



Building block types

Two types of functional units:

- Elements that operate on data values (combinational)
  - Output is function of current input
  - No memory
- Elements that contain state (sequential)
  - Output is function of current and previous inputs
  - State = memory

Combinational circuit examples

- Gates: and, or, nand, nor, xor, inverter
- Multiplexer
- Decoder
- Adder, subtractor, comparator
- ALU
- Array multipliers

Sequential circuit examples

- Flip-flops
- Counters
- Registers
- Register files
- Memories

Thanks