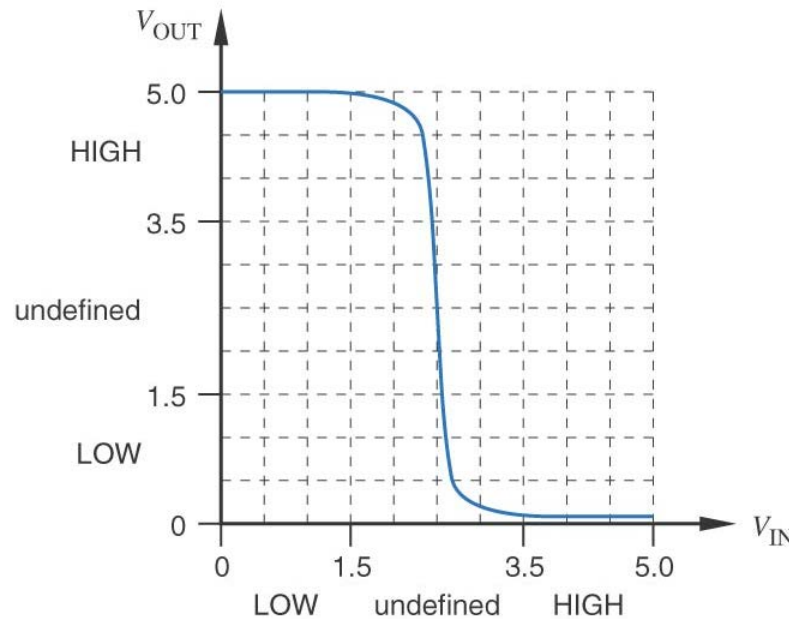


# L12 (7.1 – 7.15)

- Schmitt-trigger input
- Programmable logic array
- Fixed-point and floating-point numbers

# Digital Circuits Characteristics




Not  
guaranteed

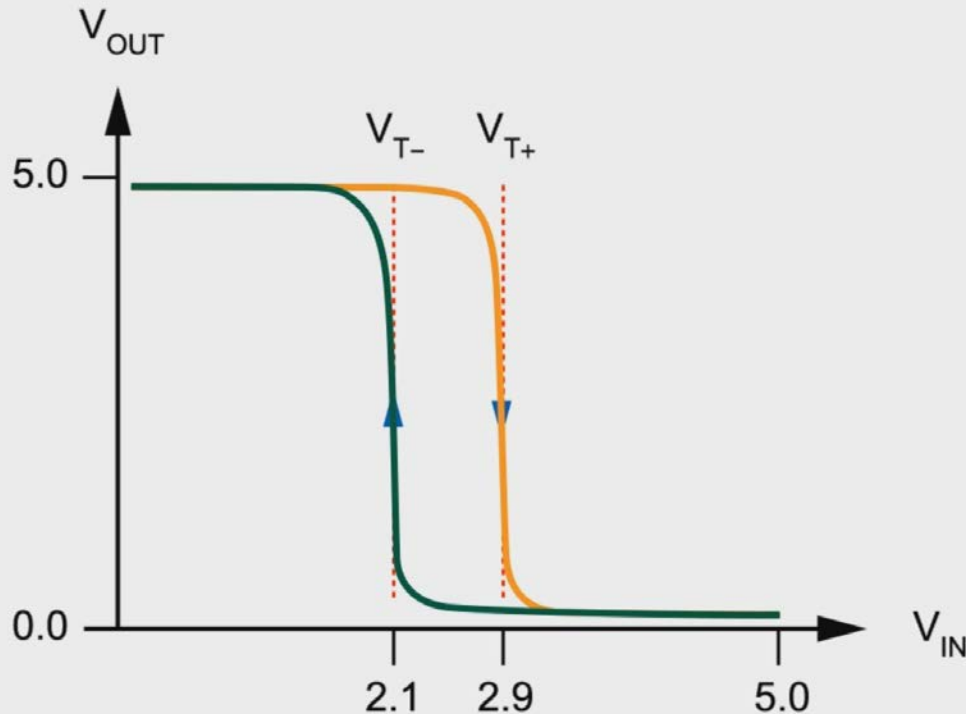
**Fig. 3-25** Typical input-output transfer characteristic of a CMOS inverter

# Schmitt-Trigger Inputs

Fig. 3-47

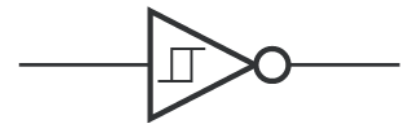
- A Schmitt-Trigger inverter is characterised by 2 threshold voltages:  $V_{T-}$  and  $V_{T+}$
- When input voltage  $V_{IN}$  rises above  $V_{T+}$ , output  $V_{OUT}$  will switch to low.
- When input voltage  $V_{IN}$  drops below  $V_{T-}$ , output  $V_{OUT}$  will switch to high.
- This non-symmetric behaviour is called **hysteresis**.
- The symbol  indicates Schmitt-Trigger input.

# Schmitt-Trigger Inverter



(a) input-output transfer characteristic

(b)

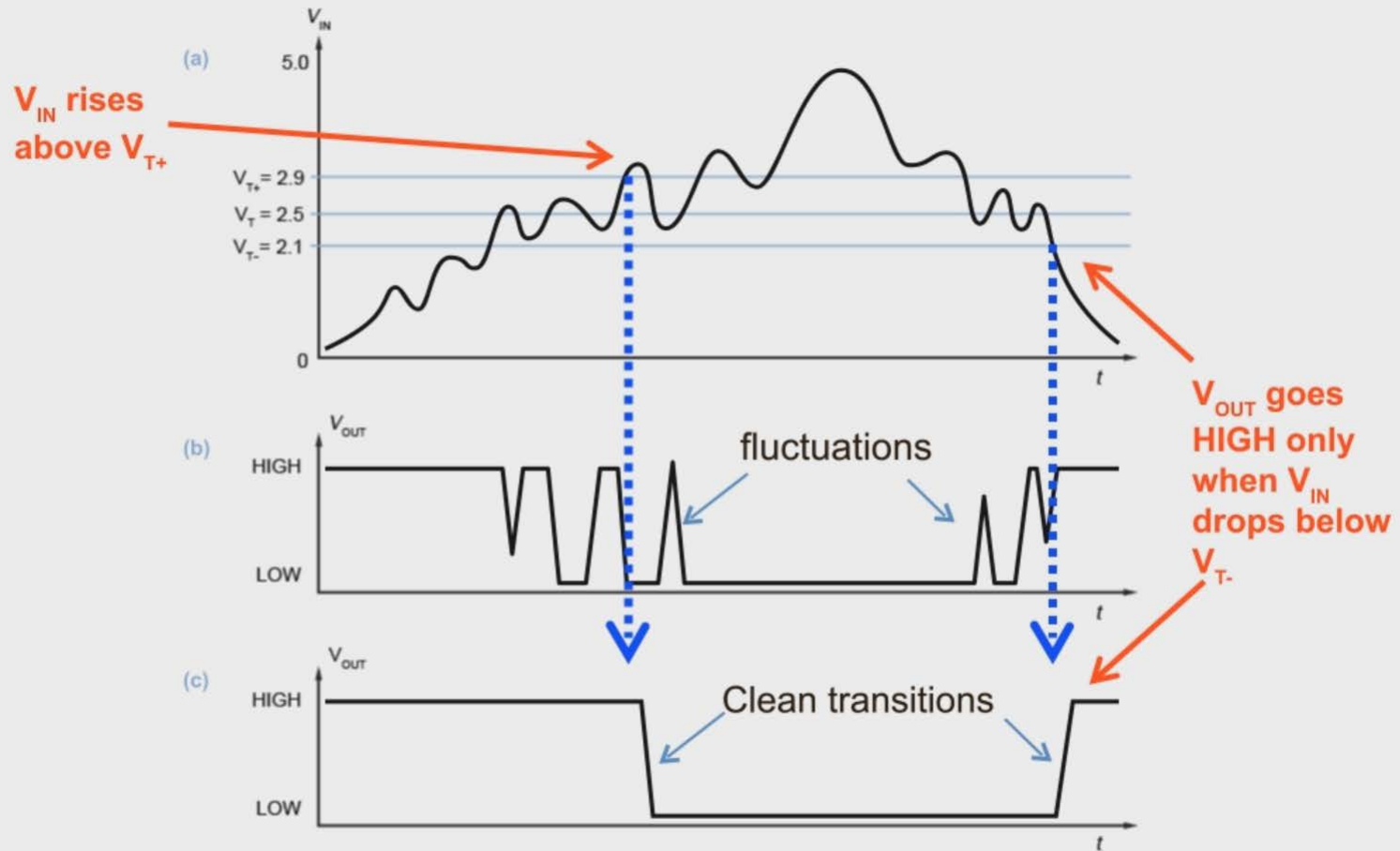


(b) logic symbol

**Fig. 3-47 Schmitt-trigger inverter**

## Schmitt-Trigger Inverter

- **Fig. 3-48**
- **When input voltage ( $V_{IN}$ ) rises slowly, or has minor fluctuations around  $V_T$ , a standard inverter will give unstable, fluctuating output.**
- **The output of a Schmitt-Trigger inverter in such situations has clean transitions and does not fluctuate.**



**Fig. 3-48 Device operation with slowly changing inputs: (a) a noisy, slowly changing input; (b) output produced by an ordinary inverter; (c) output produced by an inverter with 0.8V of hysteresis.**

# Combinational PLA

- **PLD = Programmable Logic Devices**
- **Examples are PLA, PAL, CPLD, FPGA**
- **PLA: Programmable Logic Array**
- **PAL: Programmable Array Logic**
- **CPLD: Complex PLD**
- **FPGA: Field Programmable Gate Array**
- **SOM/SOP Boolean expressions can be easily implemented on a PLA**

# PLA Example

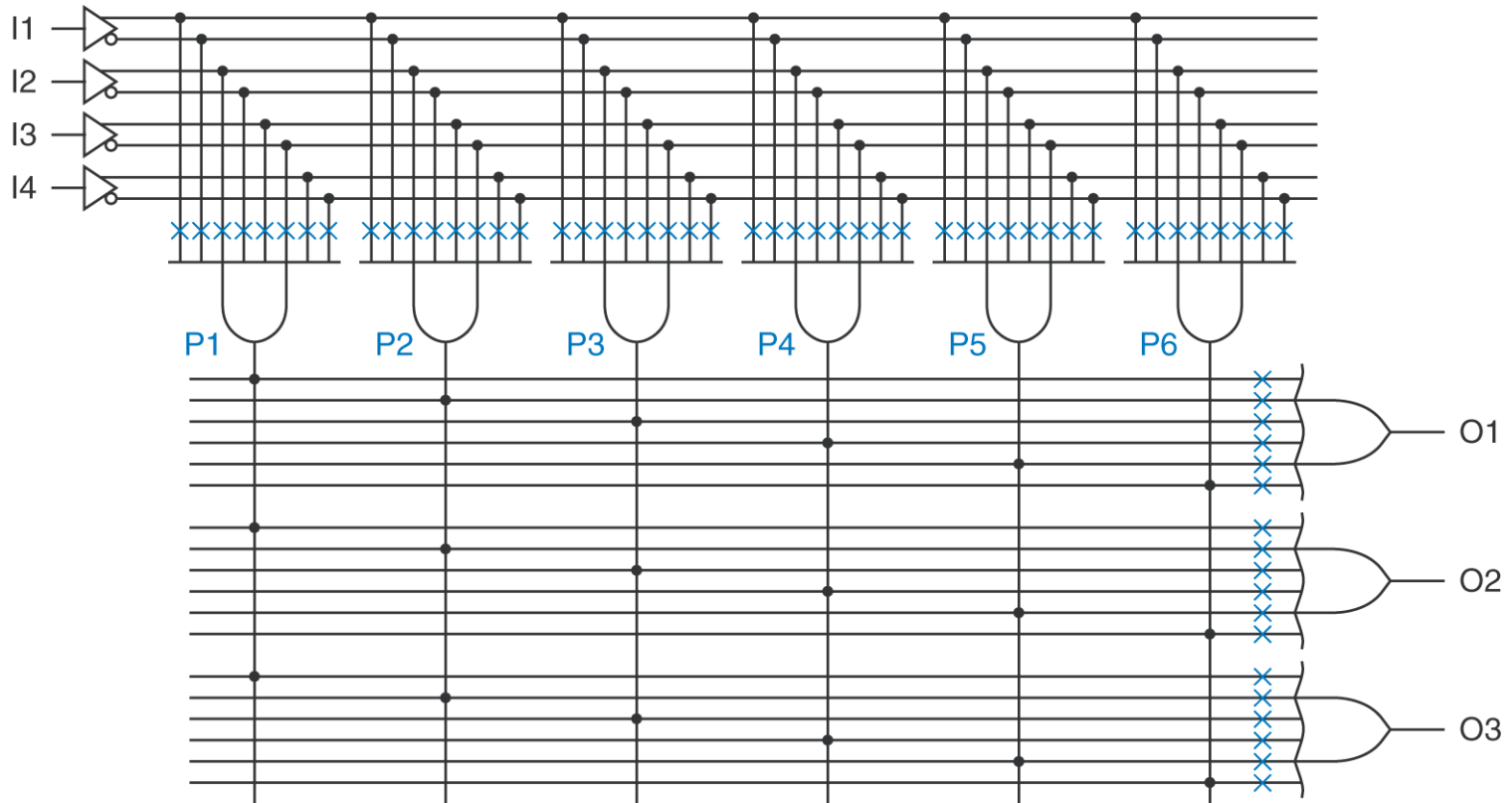


Figure 6-21

A  $4 \times 3$  PLA with six product terms.



**X : indicate programmable connections**

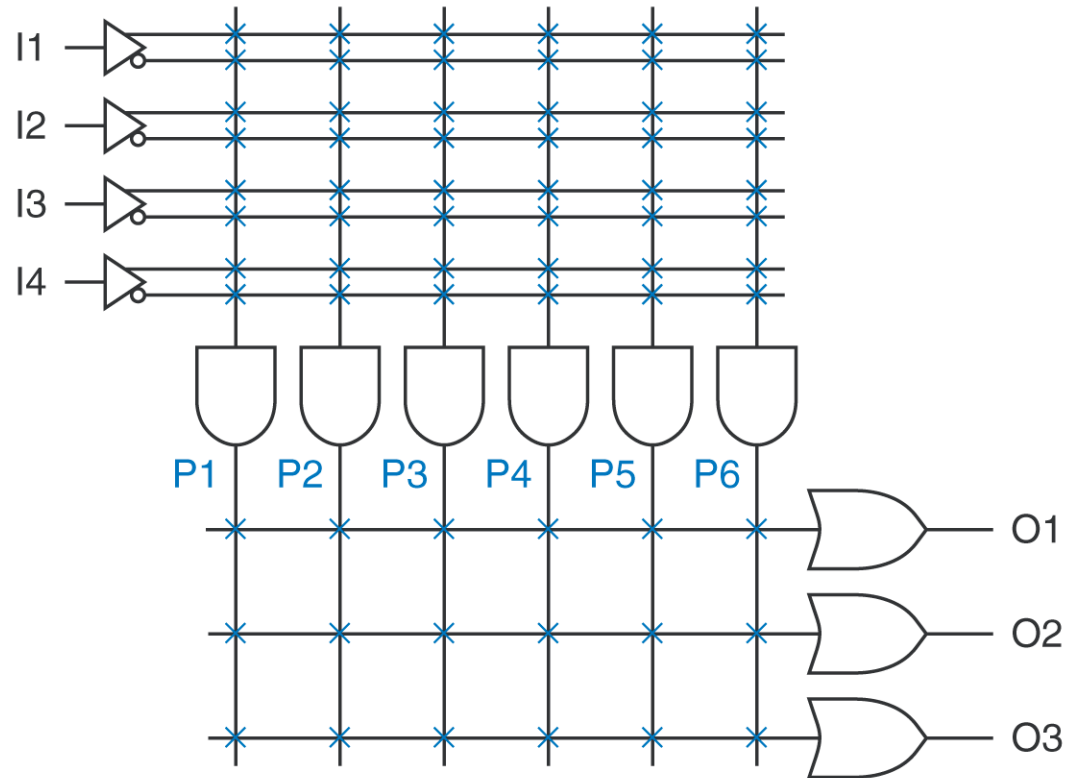


Figure 6-22

Compact representation of a  $4 \times 3$  PLA with six product terms.

# PLA Example: SOP implementation

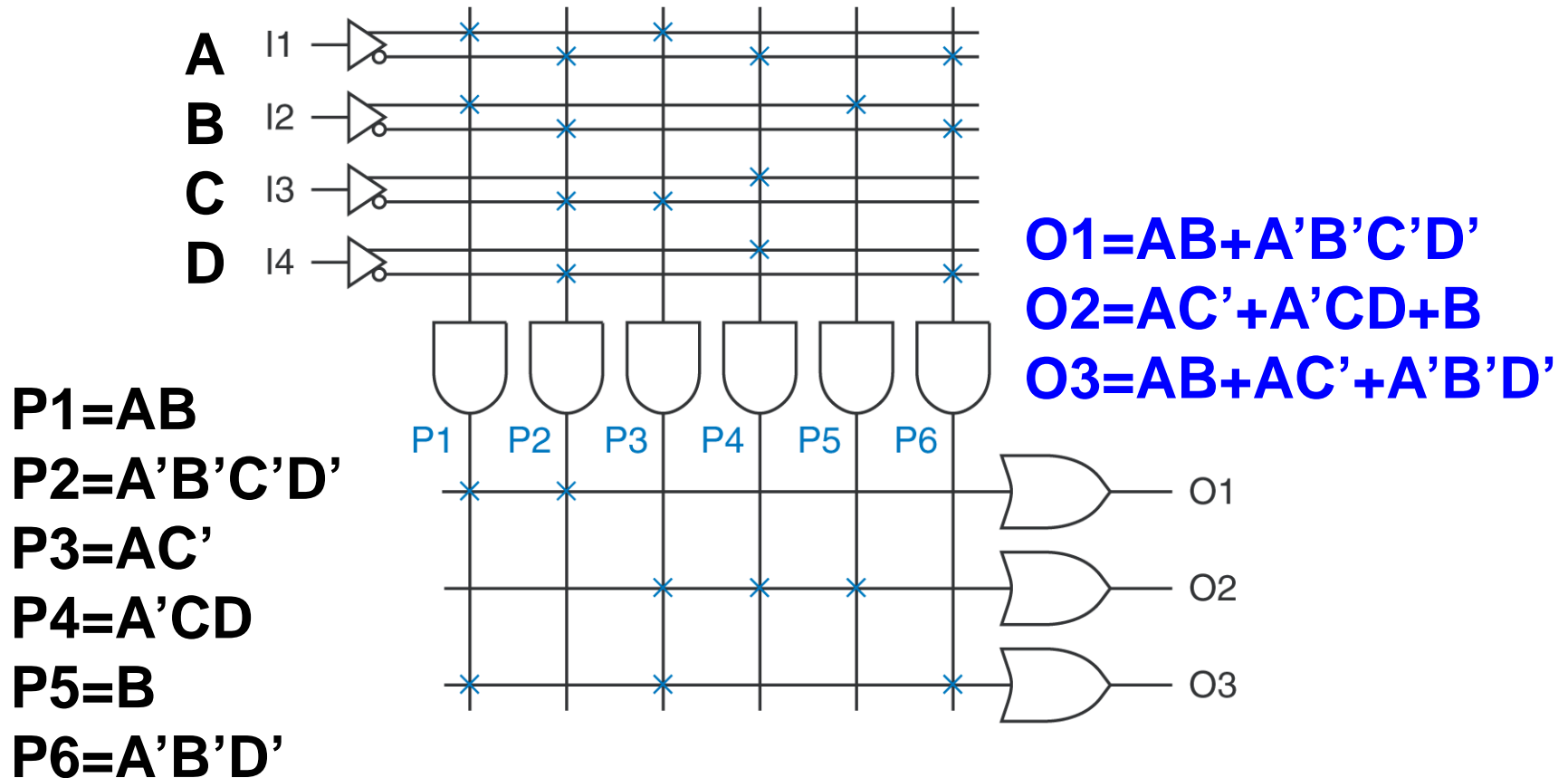


Figure 6-23

A 4 × 3 PLA programmed with a set of three logic equations.

## PLA Example (truth table)

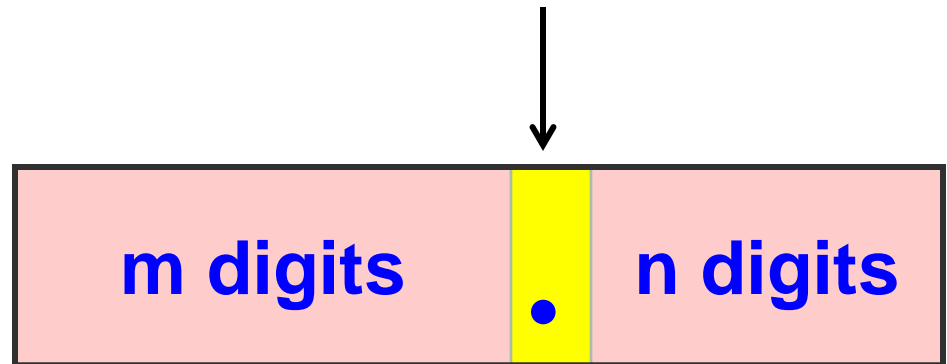
A	B	C	D	O1	O2	O3
0	0	0	0	1	0	1
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

# Fixed-point numbers

## Examples:

- $102_{10}$
- $123.456_{10}$
- $2000.AD_{16}$
- $1001.0101_2$

The radix point has a fixed position among the integer and fraction digits



- **Fixed-point numbers are usually used for representing integers only**
- **Disadvantages of using fixed-point arithmetic for non-integers:**
  - **relatively small range of numbers that can be represented**
  - **possible loss of significant digits during computation**

# Floating-point numbers

**Machine equivalent of scientific notation**

**A floating-point number  $Z$  is represented as:**

$$S \times R^E$$

**e.g.  $2.99792458 \times 10^5$**

$$S \times R^E$$

Where

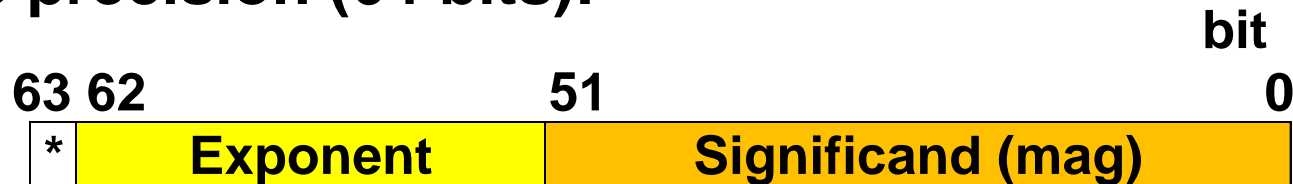
- S is the *significand*, usually a signed fractional fixed-point number
- R is the *radix* (e.g. 2 for binary, 8 for octal, 10 for decimal, etc.)
- E is the *exponent*, usually a signed fixed-point integer

# Example: the IEEE Standard 754 on floating-point arithmetic

Single precision (32 bits):



Double precision (64 bits):



- \* is the sign bit of significand
- radix is 2
- significand is in sign-magnitude notation
- exponent is in 2's complement notation