Digital Circuits

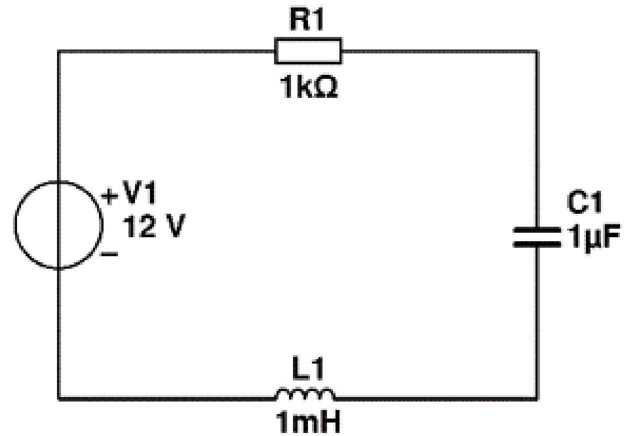
Networks and Embedded Systems

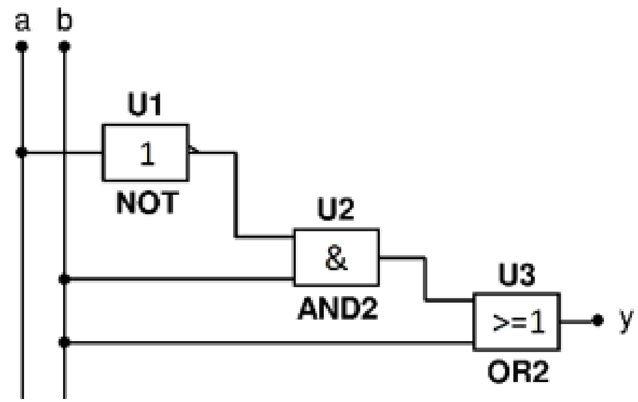
First Grade Level

Wolfgang Neff

Digital Circuits (1)

- Analog and Digital
  - Analog Circuits
    - Any voltage level allowed
  - Digital Circuits
    - Two voltage levels allowed
      - + and –
      - 1 and 0
      - H and L



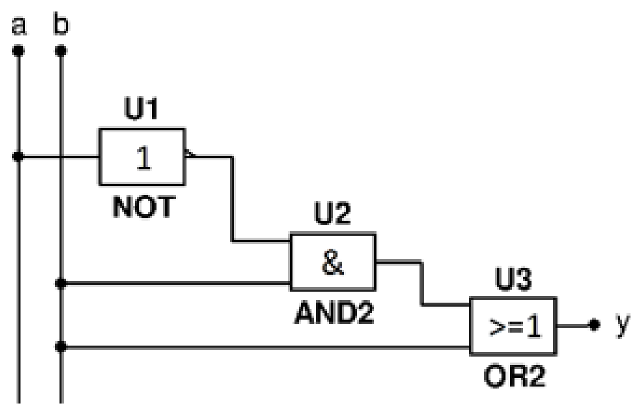


Digital Circuits (2)

- Basic Concept
  - Based on Boolean algebra
    - 0 → Low voltage
    - 1 → High voltage
    - Operator → Symbol
    - Function → Circuit
  - Terms
    - Logical functions → Switching function
    - Truth table → Switching table

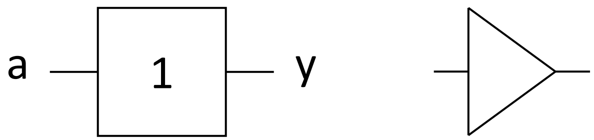
Digital Circuits (3)

- Basic Concept (continued)
  - Truth function …
    - $\varphi(a,b) = (\neg a \wedge b) \vee b$
  - … represented graphically



Digital Circuits (4)

- Graphic Symbols
  - Buffer

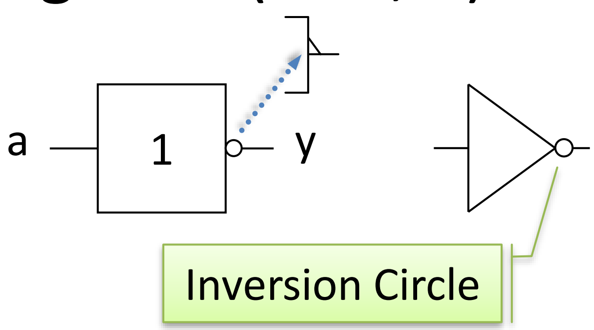


European Style (IEC 60617-12)

American Style (IEEE 315-1975)

a	y=a
0	0
1	1

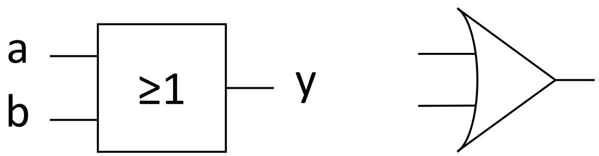
  - Negation (NOT, ¬)



a	y=¬a
0	1
1	0

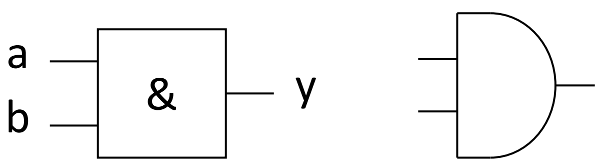
Digital Circuits (5)

- Graphic Symbols (continued)
  - Disjunction (OR, ∨)



a	b	y=a∨b
0	0	0
0	1	1
1	0	1
1	1	1

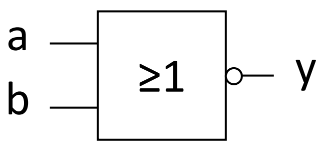
  - Conjunction (AND, ∧)

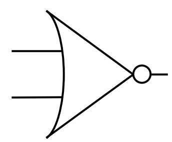


a	b	y=a∧b
0	0	0
0	1	0
1	0	0
1	1	1

Digital Circuits (6)

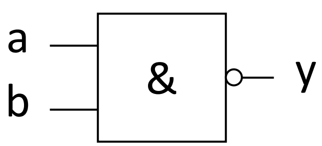
- Graphic Symbols (continued)
  - NOR (↓)

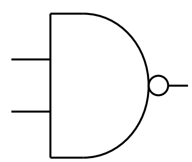




a	b	y=a↓b
0	0	1
0	1	0
1	0	0
1	1	0

  - NAND (|)

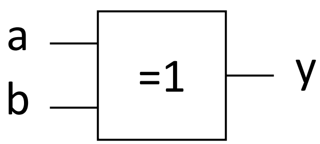


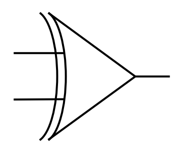


a	b	y=a|b
0	0	1
0	1	1
1	0	1
1	1	0

Digital Circuits (7)

- Graphic Symbols (finished)
  - XOR ($\oplus$)

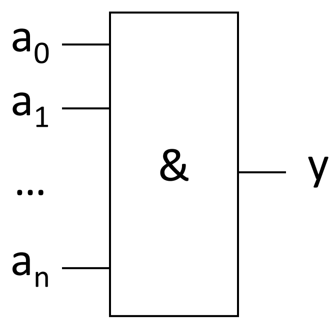




a	b	y=a$\oplus$b
0	0	0
0	1	1
1	0	1
1	1	0

- Compound Gates
  - AND-Gate
    - $y = a_0 \wedge a_1 \wedge a_2 \ldots$

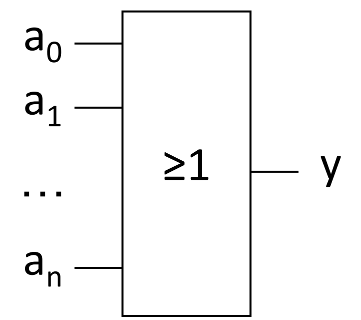
$$(a_0, a_1, \cdots) \mapsto \begin{cases} 1 \text{ if each } a_i = 1 \\ 0 \text{ else} \end{cases}$$



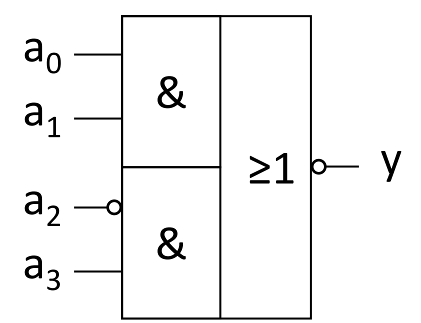
Digital Circuits (8)

- Compound Gates (continued)
  - OR-Gate
    - y = $a_0 \vee a_1 \vee a_2 \ldots$

$$(a_0, a_1, \cdots) \mapsto \begin{cases} 0 \text{ if each } a_i = 0 \\ 1 \text{ else} \end{cases}$$

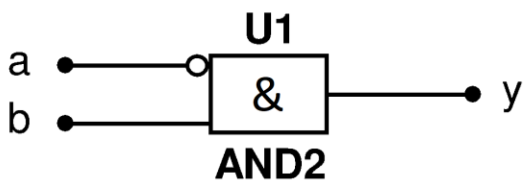


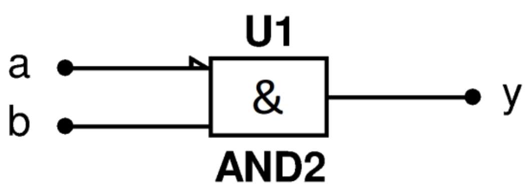
- Composition
  - Example
    - $\neg((a_0 \wedge a_1) \vee (\neg a_2 \wedge a_3))$

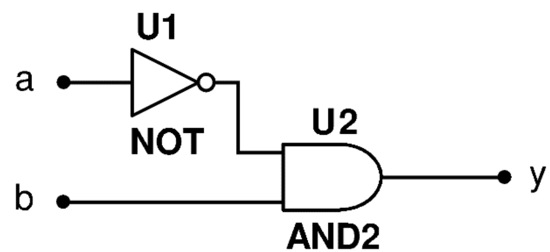


Digital Circuits (9)

- Building Blocks
  - y = ¬a ∧ b

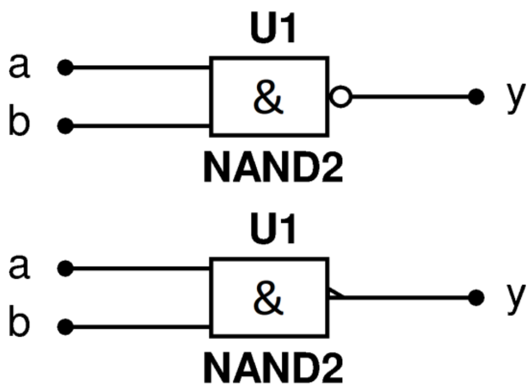


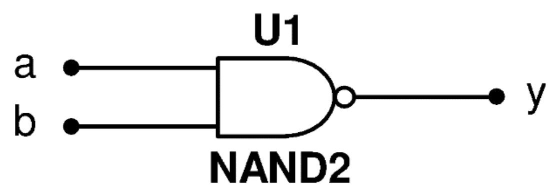




Digital Circuits (10)

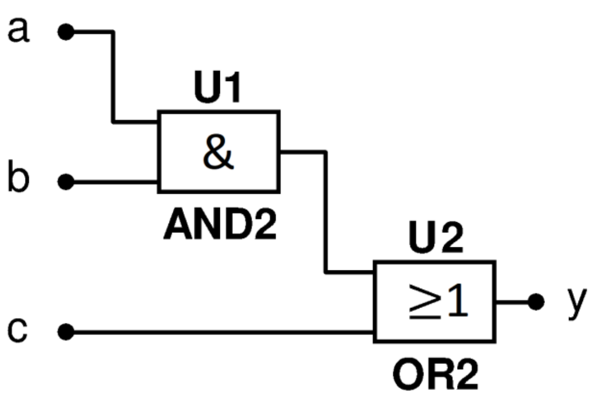
- Building Blocks (continued)
  - y = ¬(a ∧ b)

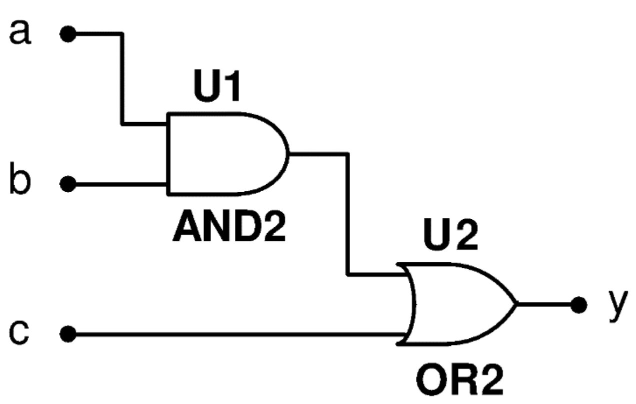




Digital Circuits (11)

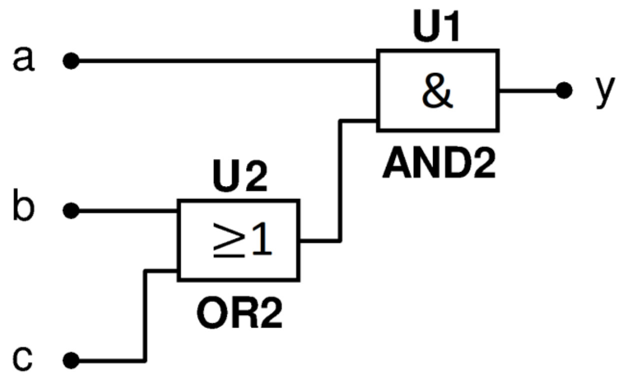
- Building Blocks (continued)
  - y = (a∧b) ∨ c

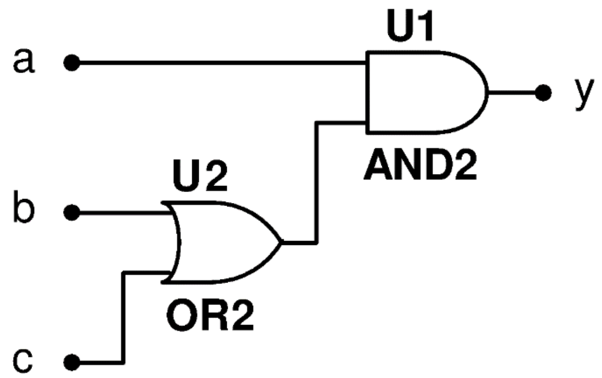




Digital Circuits (12)

- Building Blocks (finished)
  - y = a ∧ (b∨c)





Digital Circuits (13)

- Example
  - y = (a∧¬b) ∨ ¬(c∧a) ∨ c