Digital Systems I

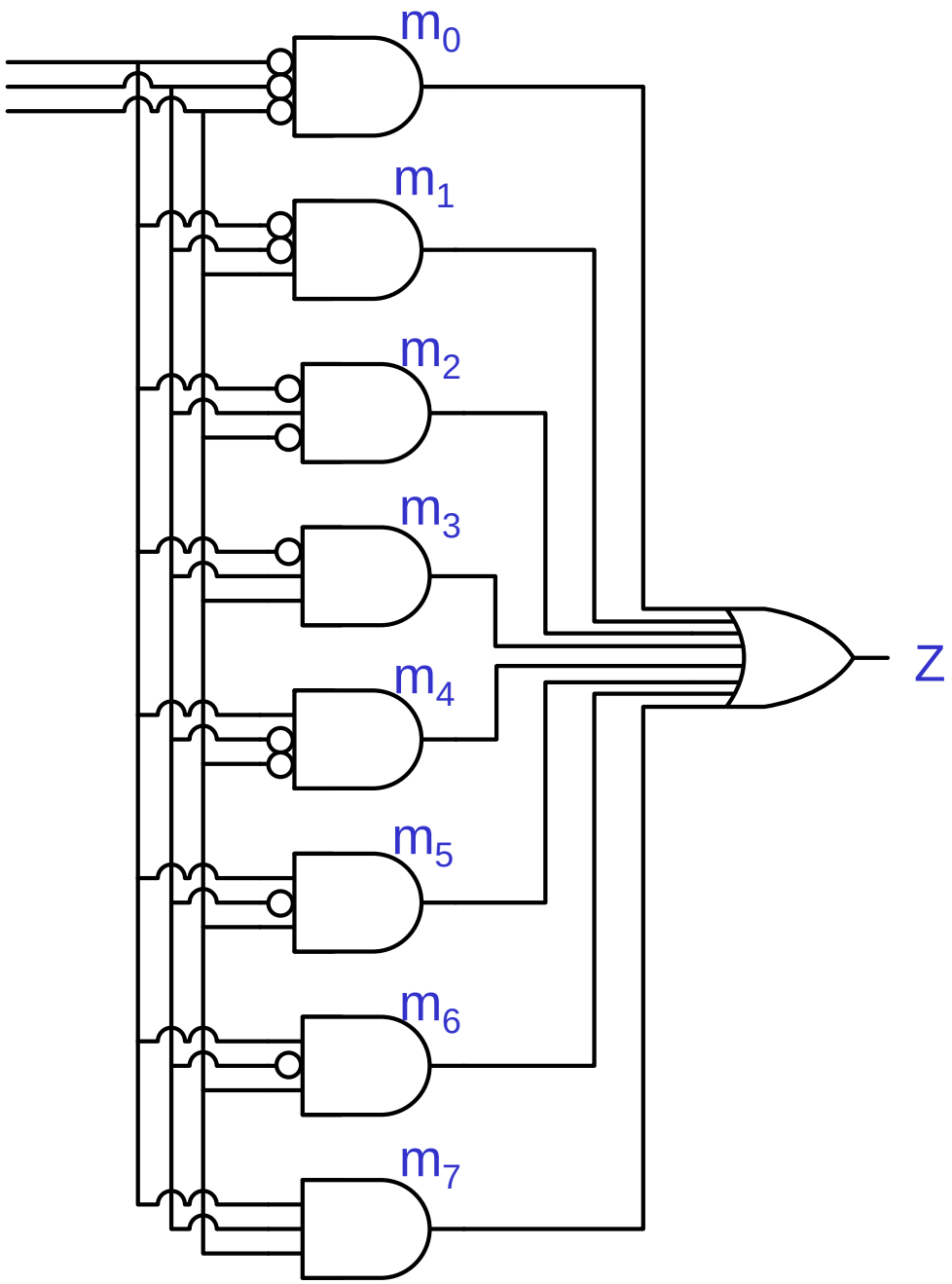
EEC 18

Lecture 4

Bevan M. Baas

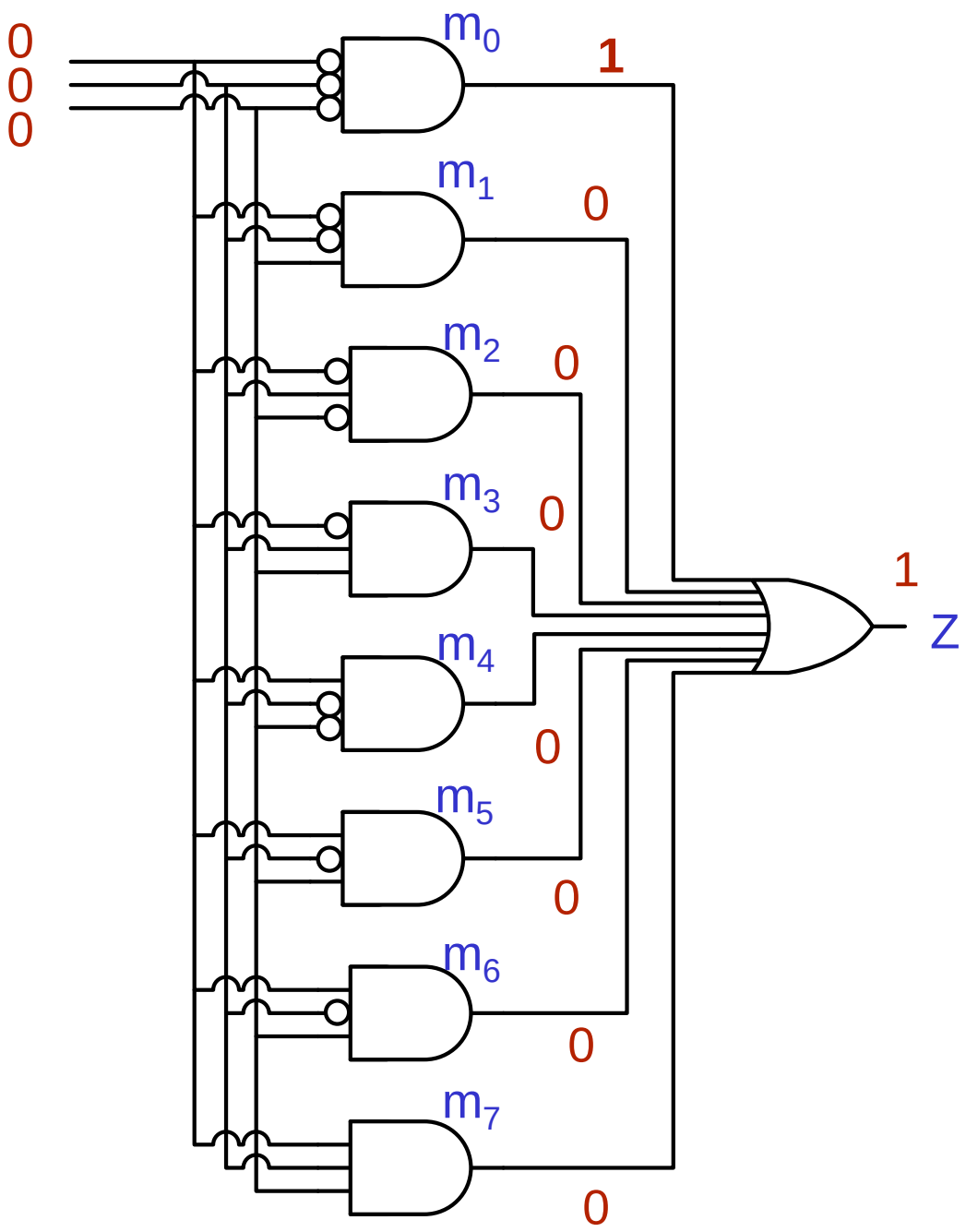
Minterm Example

- This circuit schematic shows all 8 minterms "present" for a 3-input combinational logic function
- In practice, all possible minterms would never all be present in a circuit (do you see why?)
- There is one possible minterm for each row in the truth table



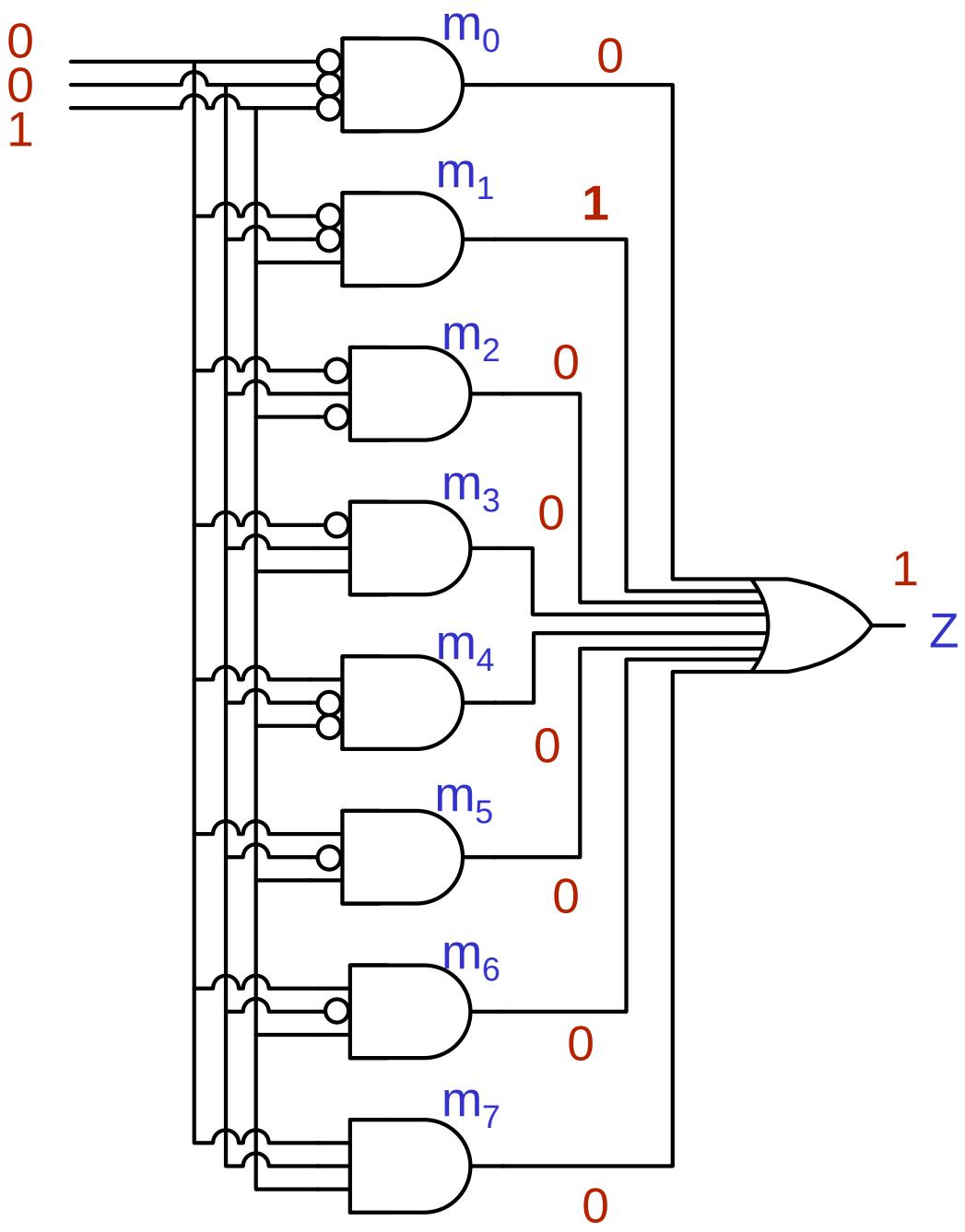
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



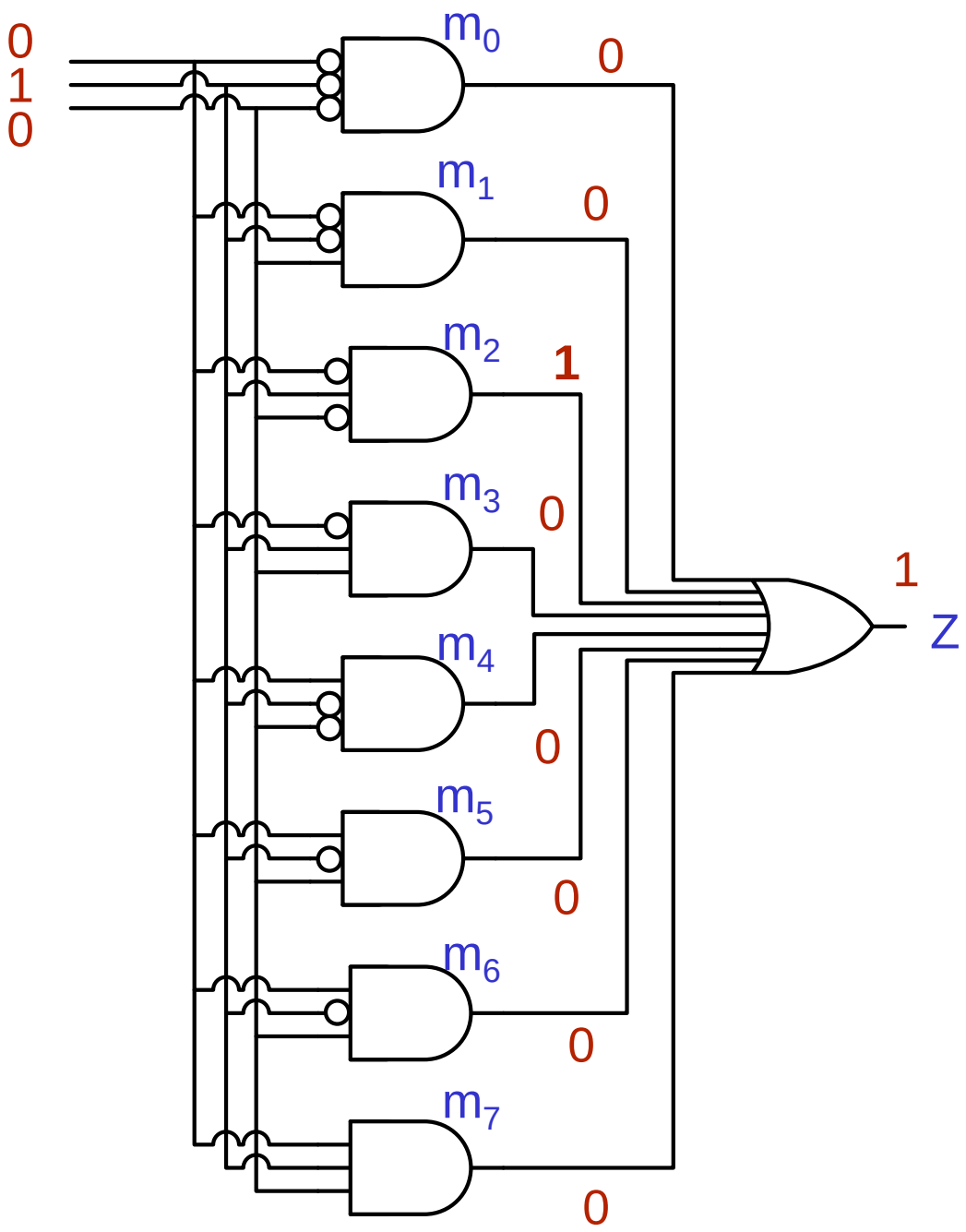
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



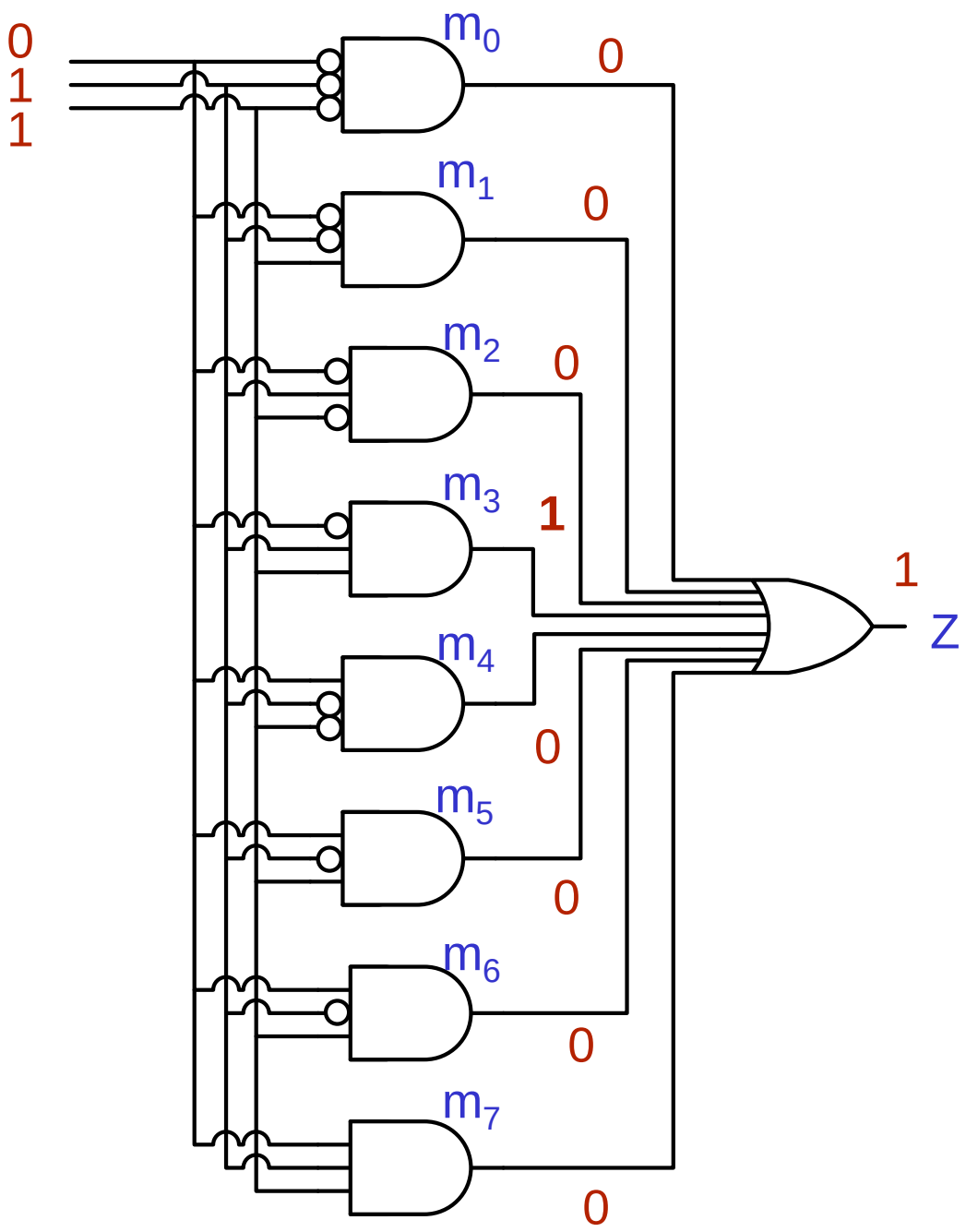
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



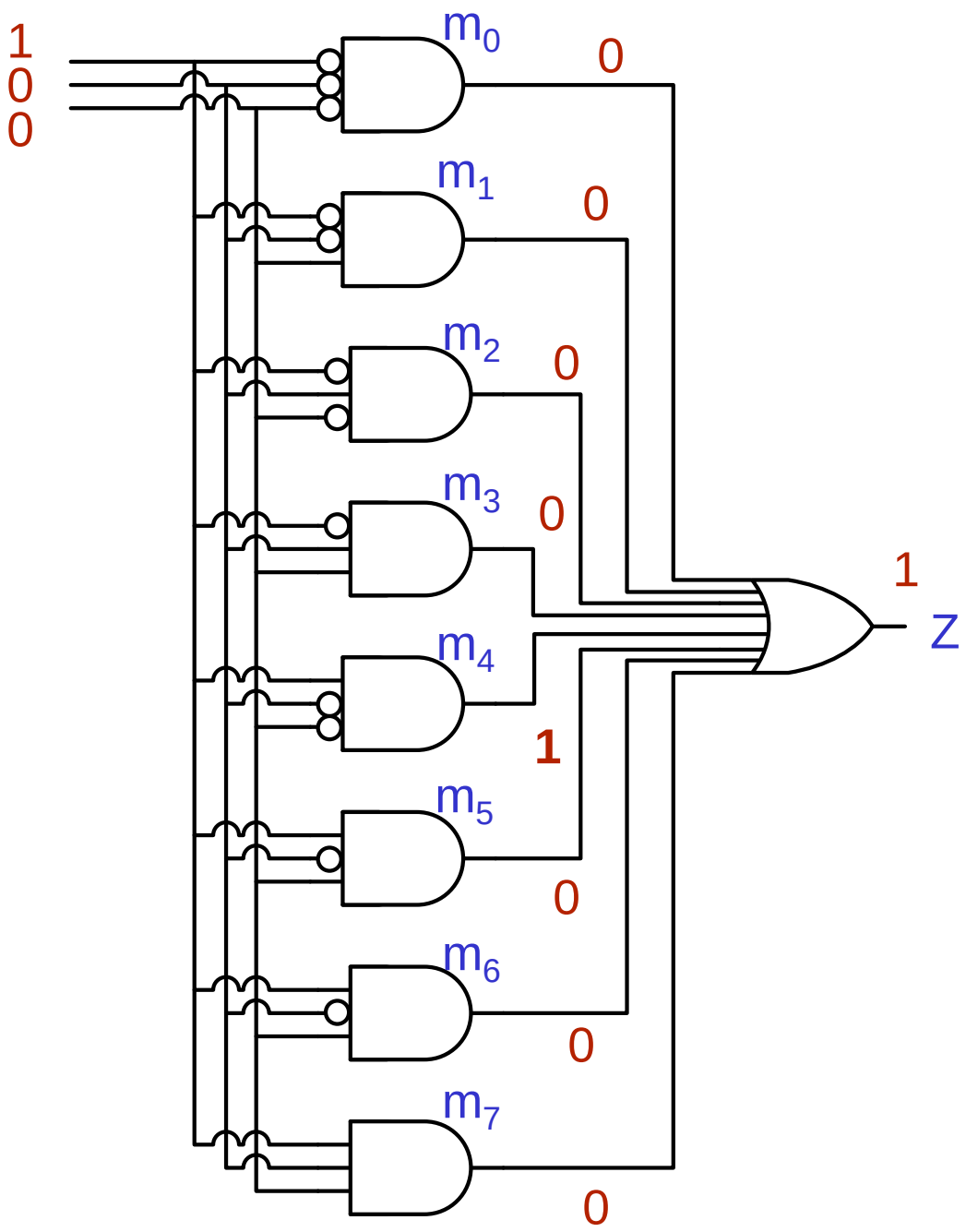
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



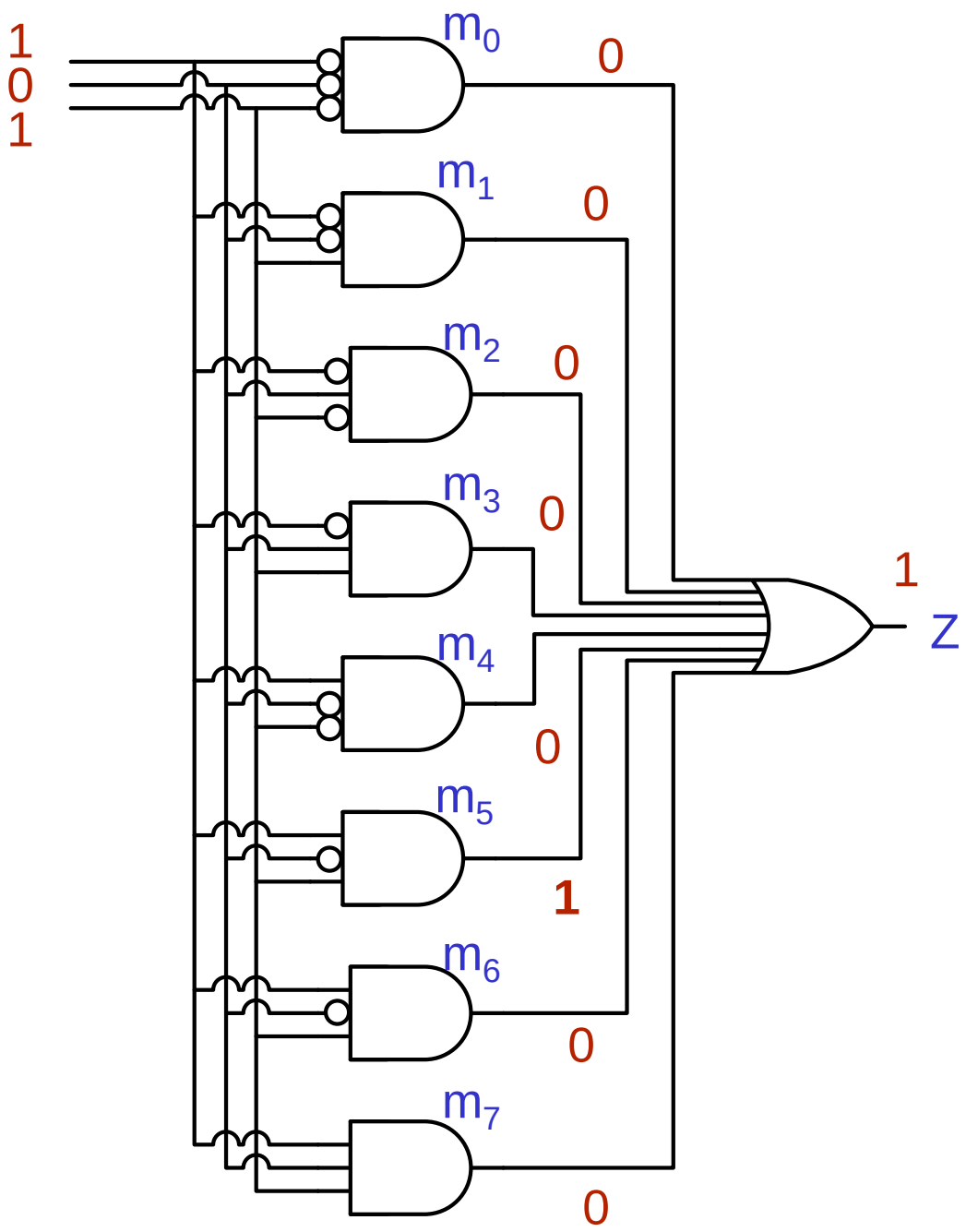
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



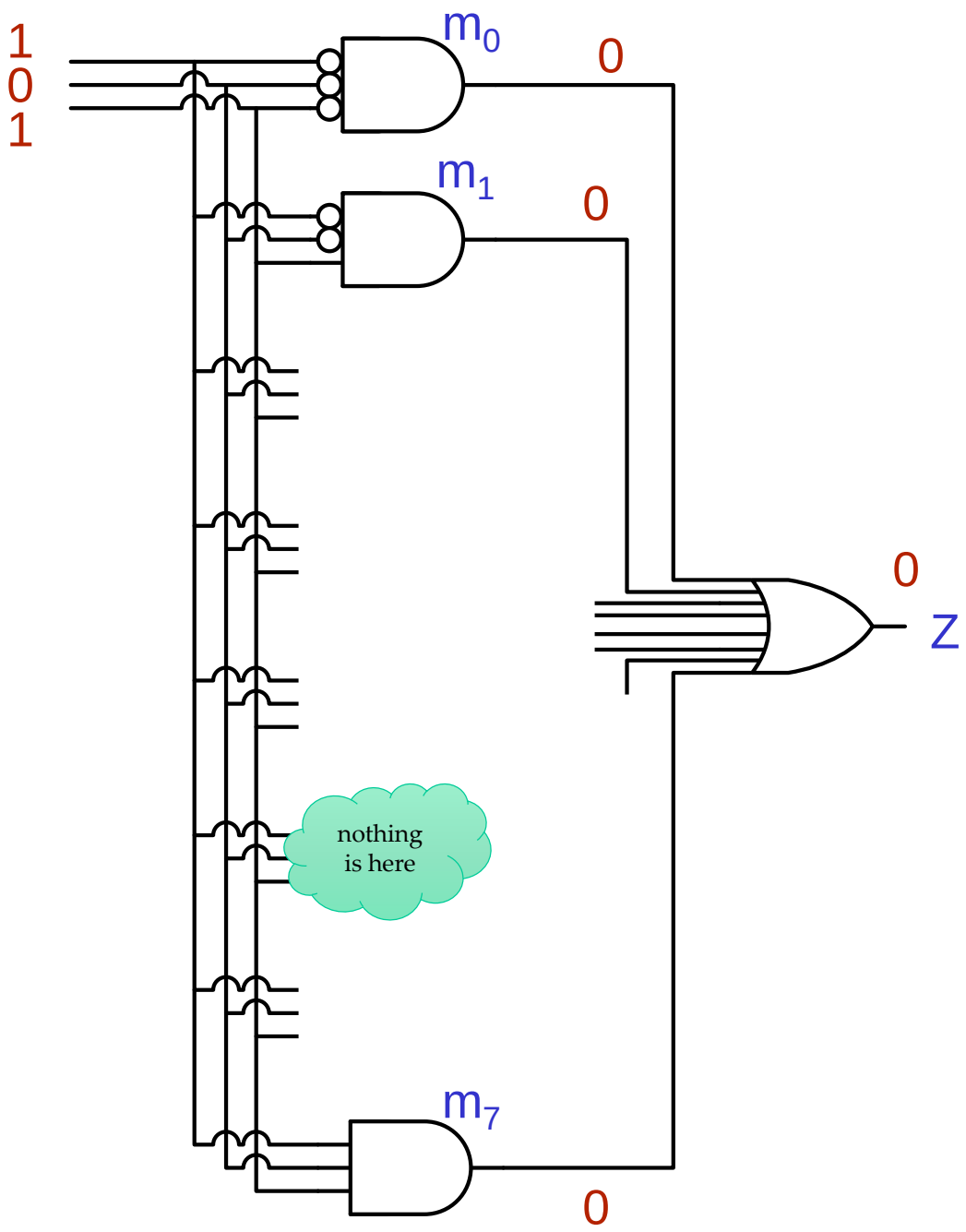
Minterm Example

- By construction, one and only one minterm is active (equals 1) at any point in time



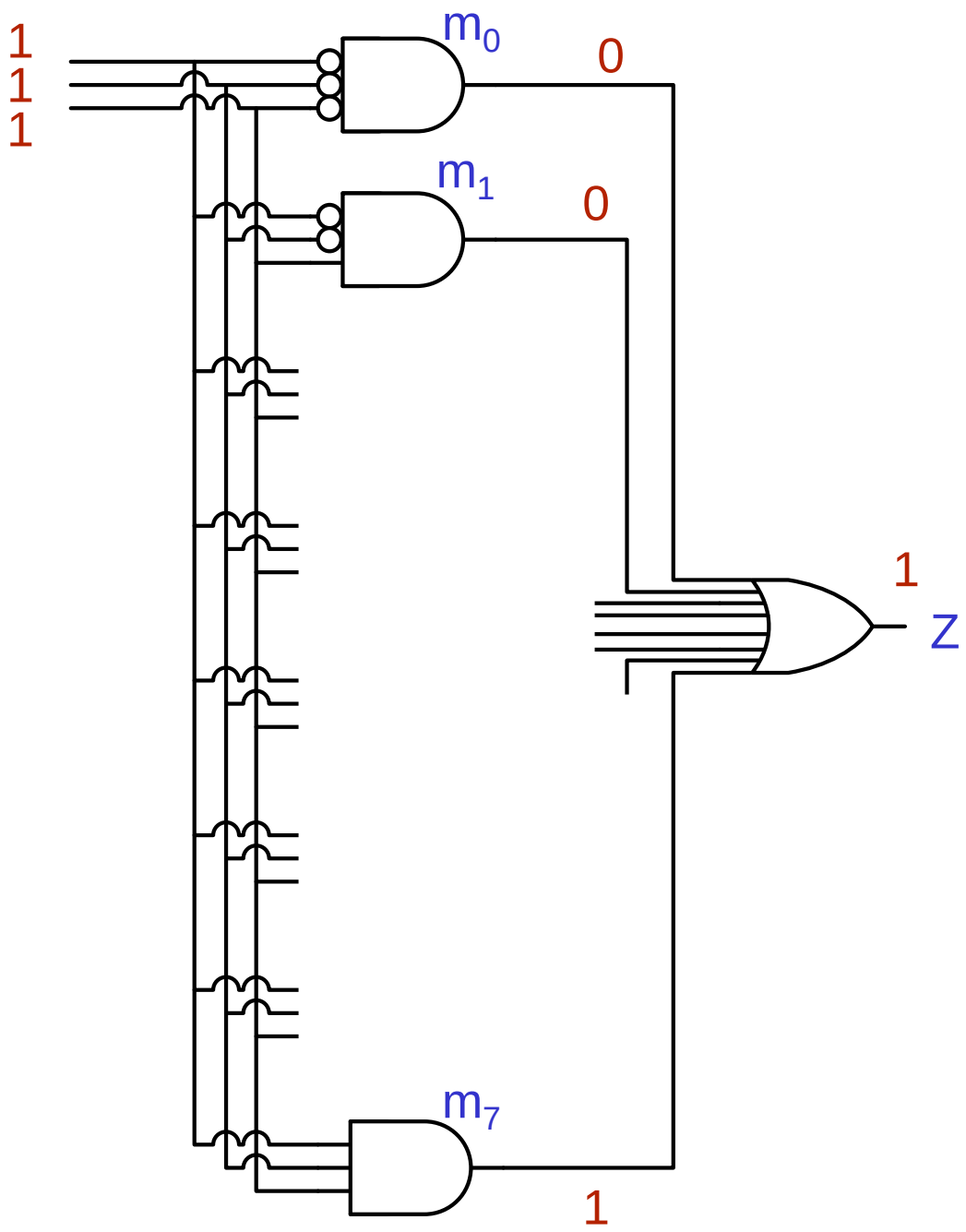
Minterm Example

- $Z = m_0 + m_1 + m_7$
- To implement an expression, a circuit is built with only the present minterm(s)
- The output can be 1 *only* when one of the present minterms forces the output to 1



Minterm Example

- $Z = m_0 + m_1 + m_7$
- To implement an expression, a circuit is built with only the present minterm(s)
- The output can be 1 *only* when one of the present minterms forces the output to 1



Minterm Example

- $Z = m_0 + m_1 + m_7$
- Of course gate inputs can not be left unconnected (unspecified). There are two solutions:
  - Tie unused inputs to a value that disables those inputs. For an OR gate, inputs would be tied to 0 (or False or Gnd)
  - The best solution is to simplify the gate. In this example, the 8-input OR gate is simplified to a 3-input OR gate