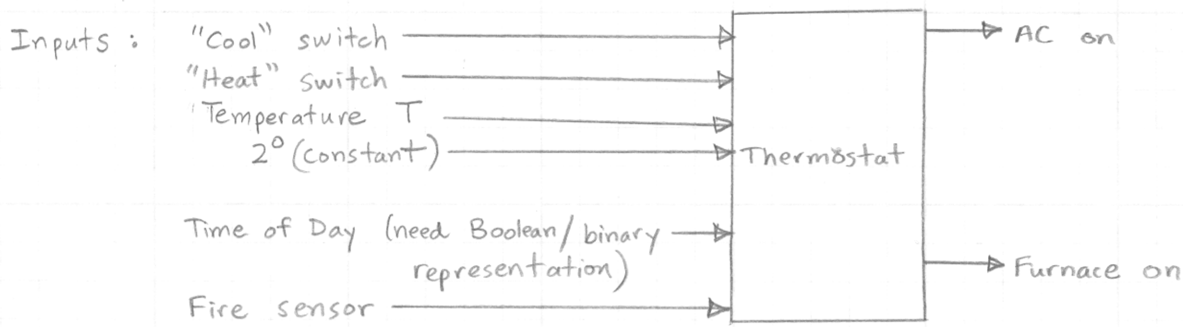
4.1 Converting Problem Statements

Three steps:

1. Figure out input(s) and output(s).
2. Understand relationship between input(s) and output(s).
3. Write an expression for each output (each output is essentially a separate problem).

Ex: Thermostat

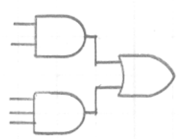
1) Air conditioner turns on if "Cool" switch is ON and temperature $T \geq 2^\circ$ above thermostat setting.
2) Furnace turns on if "Heat" switch is ON and temperature $T \leq 2^\circ$ below thermostat setting.



4.3 & 4.4 Minterm/Maxterm Expansions

We can define combinational logic circuits a number of ways:

1. Simple English description (e.g., Out = 1 if any input = 1)
2. Circuit schematic

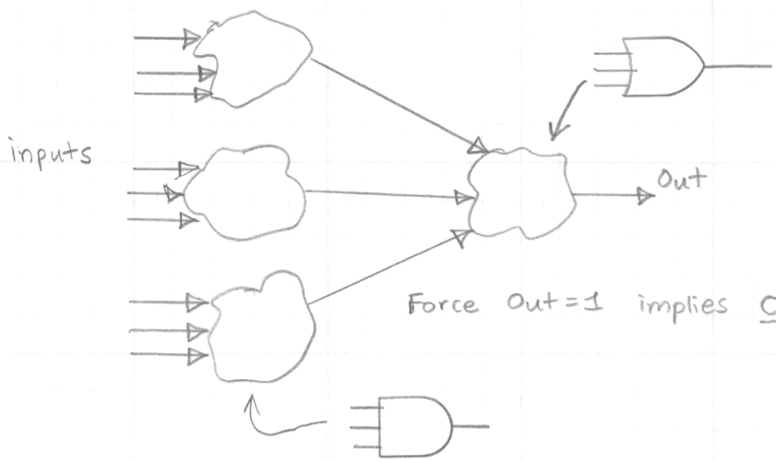


3. Truth table (enumerate all possible input combinations)
4. Minterm Expansion
5. Maxterm Expansion

(4 and 5:) New ways of describing combinational logic functions.

Minterm Expansion

Enumerate combinations of inputs which force the output to be true (1) (e.g., the "Request Stop" signal on a bus or trolley → true when anyone pulls it).



Force Out=1 implies OR operation.

Thus, a Minterm expansion is a specific Sum-of-Products expression where each entry in a truth table is a basic element (term) in the expression. These basic product terms are called minterms.
Any expression can be written/defined by its minterms.

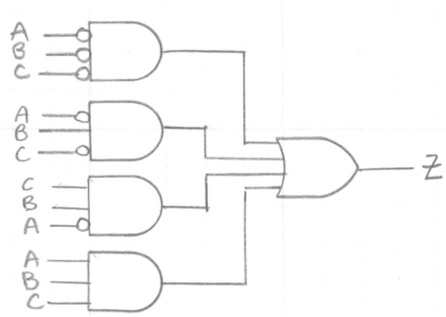
Ex: Three input truth table

A	B	C	ABC	Minterm	Z
0	0	0	000 = binary "0" = 0_{10}	m_0	1 ←
0	0	1	001 = binary "1" = 1_{10}	m_1	0
0	1	0	010 = 2_{10}	m_2	1 ←
0	1	1	011 = 3_{10}	m_3	1 ←
1	0	0	100 = 4_{10}	m_4	0
1	0	1	101 = 5_{10}	m_5	0
1	1	0	110 = 6_{10}	m_6	0
1	1	1	111 = 7_{10}	m_7	1 ←

Find an expression for Z:

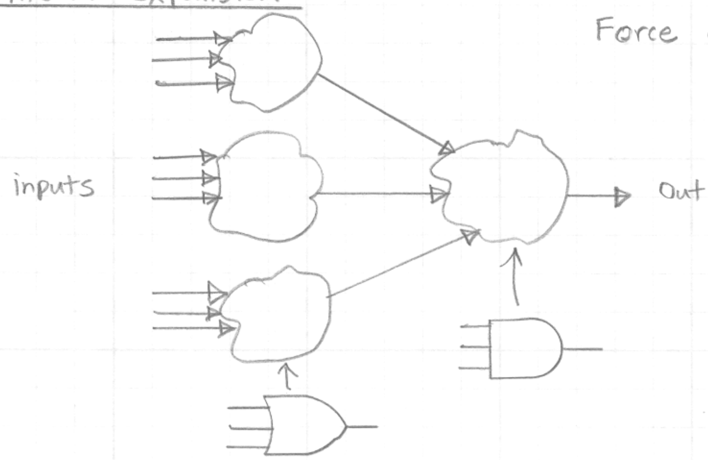
$$Z = A'B'C' + A'BC' + A'BC + ABC = m_0 + m_2 + m_3 + m_7 = \sum m(0,2,3,7)$$

Circuit implementation of Z:



Maxterm Expansion

Force Out = 0 implies AND operation.



OR output is high (1) except when all inputs are low (0).

The Maxterm Expansion is a specific Product-of-Sums expression where each each entry in a truth table is a basic term in the expression called a maxterm. Any expression can be written/defined in terms of its maxterms.

Ex: Three input truth table (same as above)

A	B	C	Maxterms	Z	
0	0	0	$A+B+C = M_0$	1	
0	0	1	$A+B+C' = M_1$	0	←
0	1	0	$A+B'+C = M_2$	1	
0	1	1	$A+B'+C' = M_3$	1	
1	0	0	$A'+B+C = M_4$	0	←
1	0	1	$A'+B+C' = M_5$	0	←
1	1	0	$A'+B'+C = M_6$	0	←
1	1	1	$A'+B'+C' = M_7$	1	

Find an expression for Z:

$$Z = (A+B+C')\cdot(A'+B+C)\cdot(A'+B+C')\cdot(A'+B'+C) = M_1\cdot M_4\cdot M_5\cdot M_6$$

$$= \prod M(1,4,5,6)$$

Note: no common terms between minterm and maxterm expansions. All 2^n (where n is the number of inputs) terms are in one or the other expression.

Minterms + Maxterms of an Inverted Expression

X	Y	Z	Z'
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

$Z = \sum m(0,1,2) = \prod M(3)$

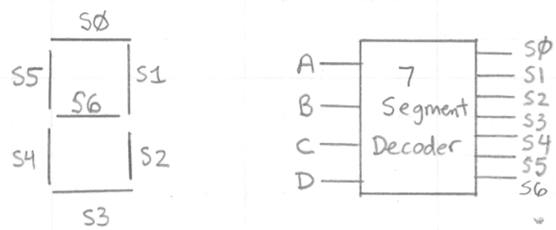
$Z' = \sum m(3) = \prod M(0,1,2)$

4.5 Incompletely Specified Functions

Sometimes we "don't care" about certain combinations of inputs in our truth table. We denote these as X's in the output columns of the truth table and we can use them to simplify expressions.

Ex: Seven Segment Display for decimal digits.

Need 4 Boolean inputs to specify the digits 0–9, but we can ignore the values $10_{10} - 15_{10}$.
← "Don't Cares"



A	B	C	D	SØ	S1	S2	S3	S4	S5	S6
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
⋮				⋮						
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
⋮				⋮						
1	1	1	1	X	X	X	X	X	X	X

In a minterm expression, denote "Don't care" as d( ).
For maxterms, use D( ).

$$S\emptyset = \sum m(0,2,\ldots,9) + \sum d(10,11,12,13,14,15)$$

$$= \prod M(1,4,\ldots) \cdot \prod D(10,11,12,13,14,15)$$