Synthesis procedure:

1. State the problem
2. Prepare TT

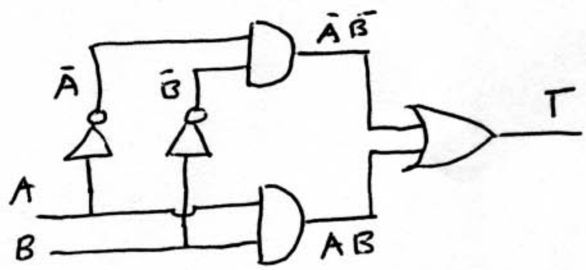
$\Big\}$ resolve any ambiguities

3. Obtain either SOP or POS canonical form
4. Optionally manipulate expression as appropriate for engineering compromises
5. Draw circuit diagram from resulting recipe.

<u>ex1</u> Simplest - very straightforward requirement:
"Design circuit to produce a logic 1 output when the two inputs are identical."

A	B	T
0	0	1
0	1	0
1	0	0
1	1	1

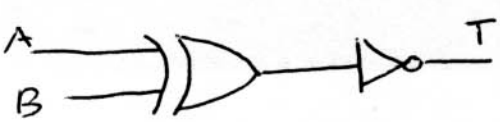
$T = \bar{A}\bar{B} + AB$ and circuit is



Reference to table of 16 functions shows this is just F_9 which is the $\overline{A \text{ XOR } B}$ fnc.

Hence first we deduce $\overline{A \text{ xor } B} = \bar{A}\bar{B} + AB$

Next we note above circuit could be built as:



which is simpler

More symbols:

A xor B is written as $A \oplus B$

$\overline{A \text{ xor } B}$ " " " $\overline{A \oplus B}$ and sometimes as $A \odot B$

Home exercise: From F_9 above show $A \oplus B = A\bar{B} + \bar{A}B$ (XOR)

Another modification possible:

$$T = \bar{A}\bar{B} + AB$$

$$\bar{T} = \overline{\bar{A}\bar{B} + AB} = \overline{\bar{A}\bar{B}} \cdot \overline{AB}$$

so $\bar{\bar{T}} = T = \overline{\overline{\bar{A}\bar{B}} \cdot \overline{AB}}$ which uses only NAND gates

(Reminder: NAND is a complete set)

<u>EX2</u> Simple word statement

"Design logic to produce a 1 output whenever any two of three inputs are 1.

X	Y	Z	T
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$T = \bar{X}YZ + X\bar{Y}Z + XY\bar{Z} + XYZ$$

uses 4 3input ANDs and 4 input OR and invertors

show this can be simplified to:

$$T = XZ + XY + YZ$$

triple 3in AND = 7411

EX3 Sometimes the word statement may use unfamiliar terms:
"Design logic to generate an odd parity bit for octal code"

Octal numbers are to base 8

e.g. $65_8 = 6 \cdot 8^1 + 5 \cdot 8^0 = 48 + 5 = 53_{10}$

↳ each digit coded in 3 bit binary

so 65_8 coded as $\underbrace{110}_{6}\ \underbrace{101}_{5}$

Parity bit added to each 3bit combination so that the total # of 1's is odd. (Could also specify even parity of course.

With these clarifications rest is straightforward

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$P = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$

and can construct circuit from this recipe. 3 OR's, 3 - 3 in AND, invertors

But note can write as:

$P = \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}C + B\bar{C})$

which should ring a bell

Bell rung but further simplification not especially obvious.

Consider: $A \oplus B = A\bar{B} + \bar{A}B$

Then

$$(A \oplus B) \oplus C = (A\bar{B} + \bar{A}B)\bar{C} + \overline{(A\bar{B} + \bar{A}B)}C$$

$$= (A\bar{B} + \bar{A}B)\bar{C} + (\bar{A} + B)(A + \bar{B})C$$

$$= A\bar{B}\bar{C} + \bar{A}B\bar{C} + \cancel{\bar{A}AC} + \bar{A}\bar{B}C + BAC + \cancel{B\bar{B}C}$$

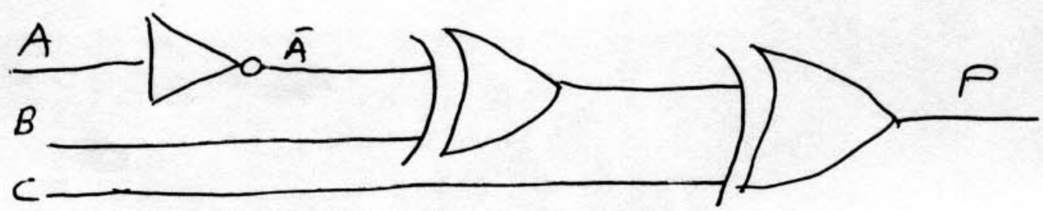
$$= A(\bar{B}\bar{C} + BC) + \bar{A}(B\bar{C} + \bar{B}C)$$

Compare to previous expression for P and deduce:

$$P = \bar{A} \oplus B \oplus C$$

home exercise: Show same result from $A \oplus (B \oplus C)$

Ckt is:



which only uses one chip (+ maybe inverter)

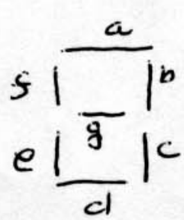
Ex: It often happens that there are multiple outputs involved in a logic design. Good example of that is a BCD to 7 segment "driver".

BCD is a very common way of representing decimal #'s. Each digit (0,1,2 ... 9) is represented in binary e.g. 9328_{10} in BCD is (Binary Coded Decimal)

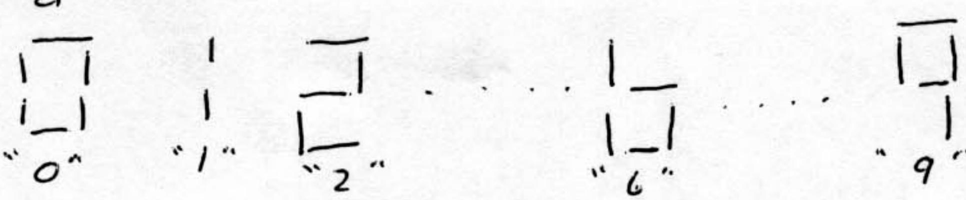
$$\underbrace{1001}_{9} \quad \underbrace{0011}_{3} \quad \underbrace{0010}_{2} \quad \underbrace{1000}_{8}$$

Note 4 binary digits (bits) are used — can accommodate 16 values but only 10 are used for BCD.

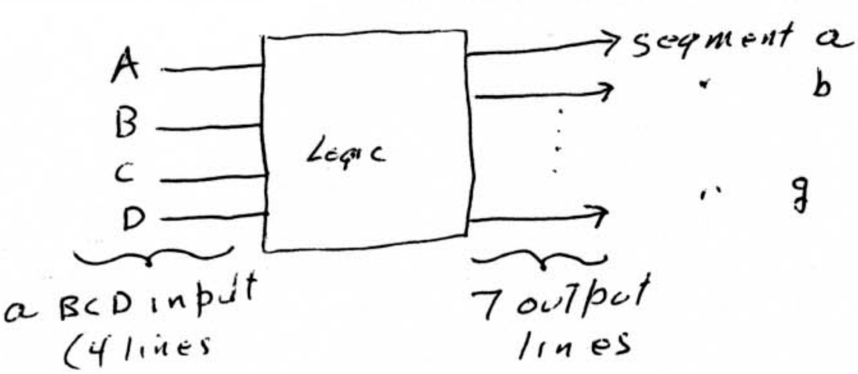
Seven segment displays are common. Each segment has an LED (light emitting diode) arranged as

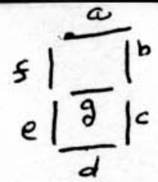


Different sets of segments are lit to represent numbers:



The driver is represented in block form as





decimal digit	BCD CODE				Display Segments							
	D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0								
5	0	1	0	1								
6	0	1	1	0	0	0	1	1	1	1	1	
7	0	1	1	1								
8	1	0	0	0								
9	1	0	0	1								

Home exercise: fill in rest of table.

Rest of design is mechanical – generate seven canonical forms and use as circuit recipe.

Fairly obvious that there will be lots of gates involved; so never build this from scratch. MSI chips exist to do the work, e.g. 7447.

Question: with 4 input variables there should be 16 rows in the T.T.: what about the others?

Don't care states:

Possible that not all entries in a TT need be included. This can ocurr e.g.

1. A priori Knowledge that certain input states will never happen such as case where only BCD inputs will be supplied to the circuit.
2. A priority ranking exists in inputs such as when a security system responds to situations which are more or less serious

Ex. Consider case where have 4 peripheral devices which generate an interrupt signal to request service & which have a priority ranking.

Assume 4 interupt signals I_0, I_1, I_2, I_3 with I_3 having highest priority .... I_0 lowest.

Need system to 1) say that an interupt signal has ocurred and 2) identify which one.

Inputs				outputs		
I_0	I_1	I_2	I_3	I	A	B
0	0	0	0	0	0	0
d	d	d	1	1	1	1
d	d	1	0	1	1	0
d	1	0	0	1	0	1
1	0	0	0	1	0	0

so: $I = I_0 + I_1 + I_2 + I_3$

$A = I_3 + I_2\bar{I}_3 = I_2 + I_3$

$B = I_3 + I_1\bar{I}_2\bar{I}_3 = I_3 + I_1\bar{I}_2$

Karnaugh Maps - an introduction

Often happens that the appropriate way to group terms to simplify algebraic expressions is not obvious. There is a visual way of summarizing things which greatly augments simplification.

Introduce by way of a design problem:

You have a summer job with Swiss AirLines working for their job recruiter. New position opens. Recruiter wants you to pre-screen applicants so that she sees only applicants who meet her specifications. You ask what are they; she replies she wants to see for further interview only those that fit:

1. Applicant is a single male of Swiss nationality
<u>or</u> 2. " is single, Swiss, and less than 25 yr old
or 3. " is female, of foreign nationality, and single
or 4. " is male and less than 25 yr old
or 5. " is single and more than 25 yr old

You have some sorting to do and some ambiguities to resolve.

Introduce variables:

Z = output =1 if candidate is accepted for further interview
=0 if rejected

A = age 1 = less than 25, 0 = over 25
B = sex 1 = male, 0 = female
C = marital status 1 = single 0 = married
D = nationality 1 = Swiss 0 = foreign

More than one way to proceed:

1. If you filled out TT you could get canonical form for Z. 16 rows to be examined.
2. Write an expression directly from criteria and then if want, convert to a canonical form.

Here can write:

$$Z = \overset{\text{single swiss male}}{C D B} + \overset{\text{single swiss under25}}{C D A} + \overset{\text{female foreign single}}{\bar{B} \bar{D} C} + \overset{\text{male under25}}{B A} + \overset{\text{single over25}}{C \bar{A}}$$

To convert to canonical note that all variables must appear in each term –

a) CDB missing A so multiply by $1 = A + \bar{A}$ to get $CDBA$, $CDB\bar{A}$

b) CDA " B " " by $1 = B + \bar{B}$ " $CDBA$, $CD\bar{B}A$

c) $\bar{B}\bar{D}C$ " A " " " $1 = A + \bar{A}$ " $\bar{B}\bar{D}CA$, $\bar{B}\bar{D}C\bar{A}$

d) BA " C and D " " " $(C+\bar{C})(D+\bar{D})$ " $ABCD$, $AB\bar{C}D$, $ABC\bar{D}$, $AB\bar{C}\bar{D}$

e) $C\bar{A}$ " B and D " " " $(B+\bar{B})(D+\bar{D})$ " $C\bar{A}BD$, $C\bar{A}\bar{B}D$, $C\bar{A}B\bar{D}$, $C\bar{A}\bar{B}\bar{D}$

Now sort out terms in RH column, drop ~~redunceres~~ redundancies and write with same order of literals:

$$Z = \underset{0011}{\bar{D}\bar{C}BA} + \underset{0100}{\bar{D}C\bar{B}\bar{A}} + \underset{0101}{\bar{D}C\bar{B}A} + \underset{0110}{\bar{D}CB\bar{A}} + \underset{0111}{\bar{D}CBA} + \underset{1011}{D\bar{C}BA}$$

$$+ \underset{1100}{DC\bar{B}\bar{A}} + \underset{1101}{DC\bar{B}A} + \underset{1110}{DCB\bar{A}} + \underset{1111}{DCBA}$$

so $Z = \sum m(3 \to 7, 11 \to 15)$ (ten terms)