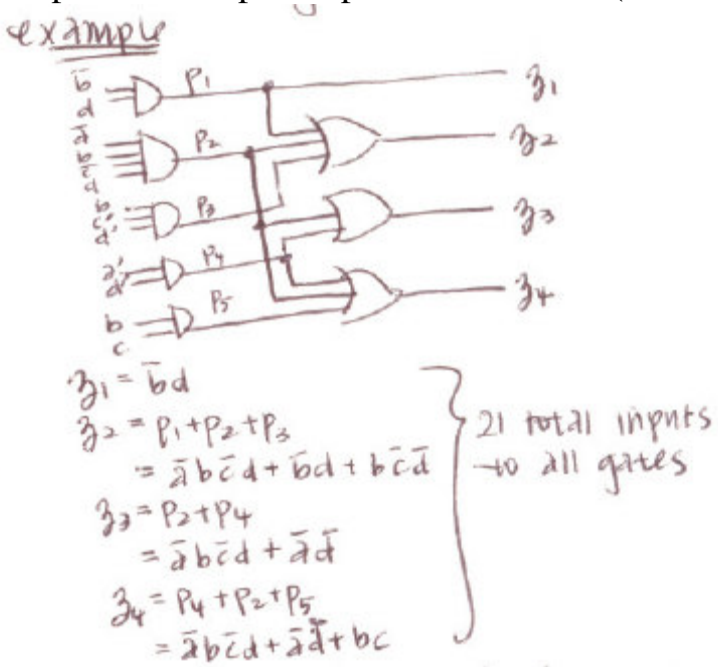
Minimization for Multiple Outputs: Quine McCluskey Method

Consider the example for multiple outputs used in class (Lecture 5)



A multiple output prime implicant of a set of functions $\{z_1, z_2, \ldots..z_m\}$ is a product of literals which is

- Either a prime implicant of one of the functions $z_k(x_1, x_2, \ldots..x_n)$ OR
- A prime implicant of one of the product functions $z_i .z_j \ldots.z_k$, where $i, j, k \leq n$

Multiple output prime implicants are the only product terms to be considered while designing a minimum cost 2-stage circuit

K-maps would contain z1, z2, z3, z4 and then z1.z2, z1.z3, z1.z4,…. which is very tedious.

Q-M method for multiple outputs

Associate ‘tags’ with every implicant. A tag is a binary word with as many bits as the number of output functions in some order. A ‘1’ implies that a

given prime implicant is a prime implicant of the output function corresponding to that position. '0' implies it is not.
So P1 (P1 implies z1 and z2, but not z3 and z4) is written as

P1 = _0_1 : 1 1 0 0
z1 z2 z3 z4

Prime Implicant Table

This is the same as before, but the tag for the combined implicants is the bitwise logical product of the tags of the implicants.
If the tag for the combined implicant is not all 0's, it contributes to atleast one output function and is entered into the table. If not, the implicant is not included in the next column. If the combined prime implicant and the original have same tags, the original can be checked off.

a b c d : Location z1 z2 z3 z4 : Tag X : Checked Off

	Column 1		**Column 2**		**Column 3**
1	0001 : 1101	X	00_1 : 1101 m1 & m3		0_ _1 : 0001
4	0100 : 0110	X	0_01 : 0101 m1 & m5		_0_1 : 1100
			_001 : 1100 m1 & m9	X	_11_ : 0001
3	0011 : 1101	X	010_ : 0110 m4 & m5		
5	0101 : 0111		_100 : 0110 m4 & m12		
6	0110 : 0001	X			
9	1001 : 1100	X	0_11 : 0001 m3 & m7	X	
12	1100 : 0110	X	_011 : 1100 m3 & m11	X	
			01_1 : 0001 m5 & m7	X	
7	0111 : 0001	X	011_ : 0001 m6 & m7	X	
11	1011 : 1100	X	_110 : 0001 m6 & m14	X	
14	1110 : 0001	X	10_1 : 1100 m9 & m11	X	
15	1111 : 0001	X	_111 : 0001 m7 & m15	X	
			111_ : 0001 m14 & m15	X	

m1 and m3 : Tags same
m1 and m5 : Tags not same. So, neither is checked off. Both are required.

01_ _ is not written, since the tag becomes 0.

Greedy algorithm (On Reduced Cover Table)

Used to obtain a minimum cost cover for a set of minterms, M using prime implicants from a set P.

- Initialize P_S to be empty
- Identify essential primes in P, P_E
- $P_S = P_E$
- Remove P_E from P and all M_E covered by P_E from M
- Search P to find subset P_L, each member of which covers largest number of members of M and not yet in P_S.
- Scan P_L to find first member P_i with fewest literals (Tie in case of same number of literals is broken by selecting first choice encountered, ie lowest subscript). Remove P_i from P and remove minterms covered by P_i from M. Add P_i to P_S.
- Repeat until M is empty. P_S is the final cover.