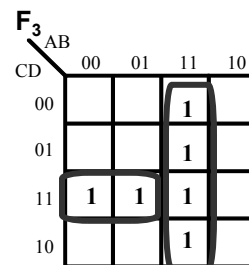
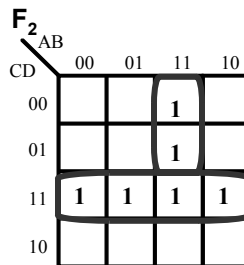
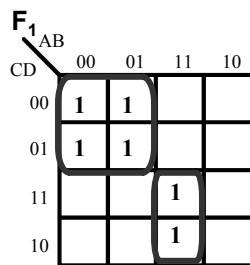


Lecture 17. Multi-output Designs

Multi-output Designs - Example

INPUT ABCD	OUTPUT abc
0000	100
0001	100
0010	000
0011	011
0100	100
0101	100
0110	000
0111	011
1000	000
1001	000
1010	000
1011	010
1100	011
1101	011
1110	101
1111	111



$$F_1 = \sum m(0, 1, 4, 5, 14, 15) = \quad +$$

$$F_2 = \sum m(3, 7, 11, 12, 13, 15) = \quad +$$

$$F_3 = \sum m(3, 7, 12, 13, 14, 15) = \quad +$$

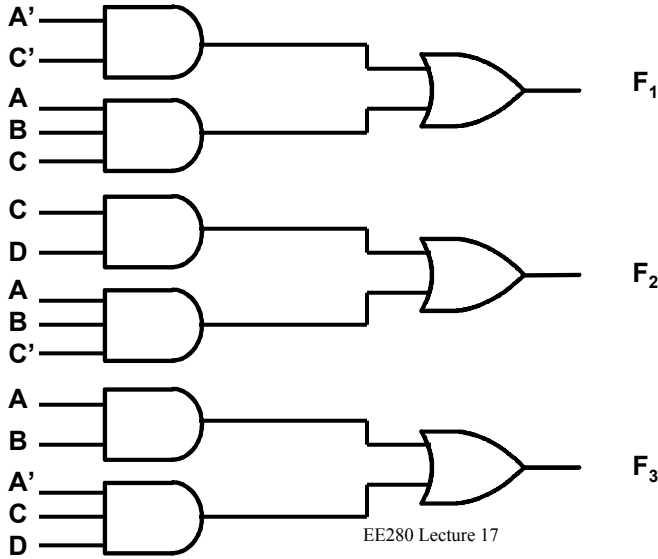
Multi-output Designs - Example

$$F_1 = A'C' + ABC$$

$$F_2 = CD + ABC'$$

$$F_3 = AB + A'CD$$

Therefore individual networks are

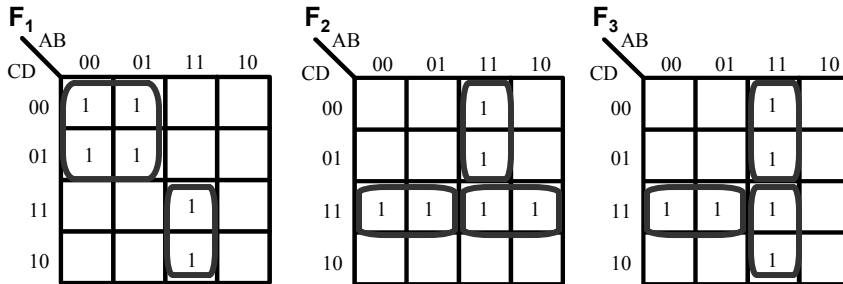


EE280 Lecture 17

17 - 3

Multi-output Designs - Example

Finding common gates



$$F_1 = \sum m(0, 1, 4, 5, 14, 15) = \quad +$$

$$F_2 = \sum m(3, 7, 11, 12, 13, 15) = \quad + \quad +$$

$$F_3 = \sum m(3, 7, 12, 13, 14, 15) = \quad + \quad +$$

EE280 Lecture 17

17 - 4

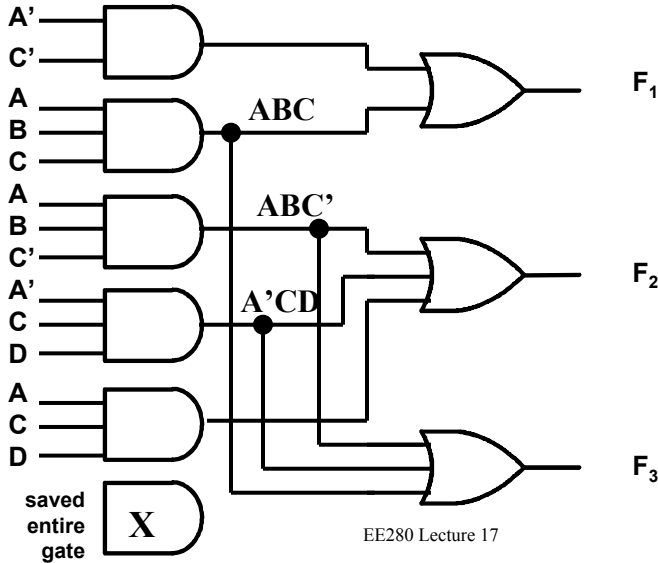
Multi-output Designs - Example

$$F_1 = A'C' + ABC$$

$$F_2 = ABC' + A'CD + ACD$$

$$F_3 = ABC' + A'CD + ABC$$

Network realization

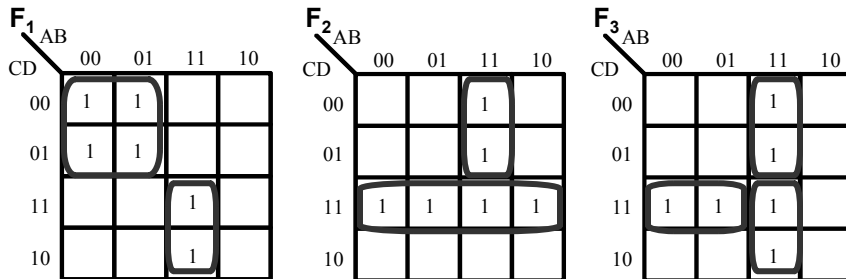


EE280 Lecture 17

17 - 5

Multi-output Designs - Example

Finding common gates (again)



$$F_1 = \sum m(0, 1, 4, 5, 14, 15) = \quad +$$

$$F_2 = \sum m(3, 7, 11, 12, 13, 15) = \quad +$$

$$F_3 = \sum m(3, 7, 12, 13, 14, 15) = \quad + \quad +$$

EE280 Lecture 17

17 - 6

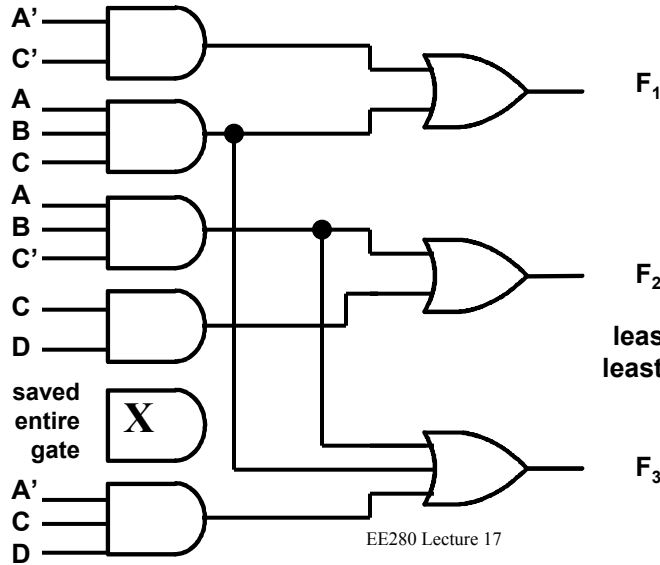
Multi-output Designs - Example

$$F_1 = A'C' + ABC$$

$$F_2 = ABC' + CD$$

$$F_3 = ABC' + AC'D + ABC$$

Network realization



least number of gates
least number of inputs
simplest circuit

EE280 Lecture 17

17 - 7

Multi-output Designs

- What we have done is retained some prime implicants, and subdivided other implicants in obtaining common terms.
- Some of the prime implicants essential to our individual functions may not be essential to the multiple-output realization.

EE280 Lecture 17

17 - 8

Multi-output Designs

"General" Procedure

1. Map and minimize each function separately. Look for common terms, which may be realized by a common gate.
Note: A term composed of a single variable connects directly to the O/P level of the network realization, and generally should not be included in any rearrangement in attempting to simplify the network.
2. Determine the prime implicants essential to single functions and the multiple-output as a whole. Such essential prime implicants should be left intact, even though there may be subsets which would be common to prime implicants in other functions.
These essential prime implicants may be determined by considering only the 1's which do not appear on any other function maps
3. Find all common prime implicants and ensure that common terms are employed as common gates in the network realization.
4. Of the terms remaining, try to use non-prime implicants in order to reduce the overall number of gates and I/P's; i.e., try to generate common terms