# CDA 3200 Digital Systems 

Instructor: Dr. Janusz Zalewski
Developed by: Dr. Dahai Guo Spring 2012

## Outline

- Combinational Logic Design Using a Truth Table
- Minterm and Maxterm Expansions
- General Minterm and Maxterm Expansions
- Incompletely Specified Functions
- Examples of Truth Table Construction
- Design of Binary Adders and Subtracters


## Combinational Logic Design Using a Truth Table (1/5)

- Sometimes, it is easier to first construct a truth table before developing the logic expression and design the logic circuit.
- The logic expression can be written in form of sum-of-products or product-ofsums, depending on how to interpret the truth table.


## Combinational Logic Design Using a Truth Table (2/5)

- Any combination of 011,100 , 101,110 or 111 can make $\mathrm{f}=1$
- ABC are $011 \rightarrow \mathrm{~A}^{\prime} \mathrm{BC}=1$
- $A B C$ are $100 \rightarrow A B^{\prime} C^{\prime}=1$
- ABC are $101 \rightarrow \mathrm{AB} \mathrm{C}^{\prime}=1$
- ABC are $110 \rightarrow \mathrm{ABC}^{\prime}=1$
- ABC are $111 \rightarrow \mathrm{ABC}=1$

| $A$ | $B$ | $C$ | $d e c$ | $f$ | $f$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 2 | 0 | 1 |
| 0 | 1 | 1 | 3 | 1 | 0 |
| 1 | 0 | 0 | 4 | 1 | 0 |
| 1 | 0 | 1 | 5 | 1 | 0 |
| 1 | 1 | 0 | 6 | 1 | 0 |
| 1 | 1 | 1 | 7 | 1 | 0 |

## Combinational Logic Design Using a Truth Table (3/5)

- Therefore, the logic expression is
$-f=A^{\prime} B C+A B^{\prime} C^{\prime}+A B^{\prime} C+A B C^{\prime}+A B C$
$-=A^{\prime} B C+A B^{\prime}+A B$
$-=A^{\prime} B C+A$
$-=\left(A^{\prime}+A\right)(A+B C)$
$-=A+B C$


## Combinational Logic Design Using a Truth Table (4/5)

- Any combination of 000,001 , or 010 can make $f^{\prime}=1$
- ABC are $000 \rightarrow \mathrm{~A}^{\prime} \mathrm{B}^{\prime} \mathrm{C}^{\prime}=1$
- $A B C$ are $001 \rightarrow A^{\prime} B^{\prime} C=1$
- $A B C$ are $010 \rightarrow A^{\prime} B C^{\prime}=1$

| $A$ | $B$ | $C$ | $d e c$ | $f$ | $f^{\prime}$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 2 | 0 | 1 |
| 0 | 1 | 1 | 3 | 1 | 0 |
| 1 | 0 | 0 | 4 | 1 | 0 |
| 1 | 0 | 1 | 5 | 1 | 0 |
| 1 | 1 | 0 | 6 | 1 | 0 |
| 1 | 1 | 1 | 7 | 1 | 0 |

## Combinational Logic Design Using a Truth Table (5/5)

- Therefore, the logic expression for $f^{\prime}$ is $-f^{\prime}=A^{\prime} B^{\prime} C^{\prime}+A^{\prime} B^{\prime} C+A^{\prime} B C^{\prime}$
$-\left(f^{\prime}\right)^{\prime}=\left(A^{\prime} B^{\prime} C^{\prime}+A^{\prime} B^{\prime} C+A^{\prime} B C^{\prime}\right)^{\prime}$
$-f=\left(A^{\prime} B^{\prime} C^{\prime}\right)^{\prime}\left(A^{\prime} B^{\prime} C\right)^{\prime}\left(A^{\prime} B C^{\prime}\right)^{\prime}$
$-=(A+B+C)\left(A+B+C^{\prime}\right)\left(A+B^{\prime}+C\right)$


## Minterm and Maxterm Expansions

 (1/10)- A literal is a variable or its complement.
- A minterm of $\boldsymbol{n}$ variables is a product of $\boldsymbol{n}$ literals in which each variable appears once in either true or complemented form, but not both.
- For a system with 3 variables
$-A B C$ is a minterm
$-A B^{\prime} C^{\prime}$ is a minterm
$-A^{\prime} C^{\prime}$ is NOT a minterm


## Minterm and Maxterm Expansions (2/10)

- A minterm is designated $\boldsymbol{m}_{\boldsymbol{i}}$, where $\boldsymbol{i}$ is the decimal value of the binary string of the variables.
- ABC
$\mathrm{m}_{7}$
- A'B'C'
$\mathrm{m}_{0}$
- ABC'
$\mathrm{m}_{6}$


## Minterm and Maxterm Expansions (3/10)

- The truth table of a logic function can be represented by a sum of minterms and in this case it is called a minterm expansion or a standard sum of products


## Minterm and Maxterm Expansions (4/10)

- $f=A^{\prime} B C+A B^{\prime} C^{\prime}+A B^{\prime} C+A B C^{\prime}+A B C$

| $A$ | $B$ | $C$ | $d e c$ | $f$ | $f$ |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 2 | 0 | 1 |  |
| 0 | 1 | 1 |  | 3 | 1 | 0 |
| 1 | 0 | 0 | 4 | 1 | 0 |  |
| 1 | 0 | 1 | 5 | 1 | 0 |  |
| 1 | 1 | 0 | 6 | 1 | 0 |  |
| 1 | 1 | 1 | 7 | 1 | 0 |  |

## Minterm and Maxterm Expansions (5/10)

- Given a truth table, if the output of a certain row is 1 , the corresponding minterm must be present in the logic expression.


## Minterm and Maxterm Expansions (6/10)

- A maxterm of $\boldsymbol{n}$ variables is a sum of $\boldsymbol{n}$ literals.
- In a system with three variables
$-(A+B+C)$ is a maxterm
$-\left(A^{\prime}+B^{\prime}+C\right)$ is a maxterm
$-\left(A^{\prime}+B\right)$ is not maxterm
- A maxterm is designated $\boldsymbol{M}_{\boldsymbol{i}}$, where $\boldsymbol{i}$ is the decimal value of the complement of the binary string.


## Minterm and Maxterm Expansions (7/10)

- $f=(A+B+C)\left(A+B+C^{\prime}\right)\left(A+B^{\prime}+C\right)$
- $f=M_{0} M_{1} M_{2}$
- $\mathrm{f}=\prod M(0,1,2)$

| $A$ | $B$ | $C$ | $d e c$ | $f$ | $f$ |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 2 | 0 | 1 |  |
| 0 | 1 | 1 | 3 | 1 | 0 |  |
| 1 | 0 | 0 | 4 | 1 | 0 |  |
| 1 | 0 | 1 | 5 | 1 | 0 |  |
| 1 | 1 | 0 | 6 | 1 | 0 |  |
| 1 | 1 | 1 | 7 | 1 | 0 |  |

## Minterm and Maxterm Expansions (8/10)

- Converting a general logic expression into a minterm expansion
- Through repeatedly applying $X+X^{\prime}=1$
$-\mathrm{f}=\mathrm{a}^{\prime} \mathrm{b}^{\prime}+\mathrm{a}^{\prime} \mathrm{d}+\mathrm{acd}{ }^{\prime}$
$-=a^{\prime} b^{\prime}\left(c+c^{\prime}\right)\left(d+d^{\prime}\right)+a^{\prime} d\left(b+b^{\prime}\right)\left(c+c^{\prime}\right)+a c d^{\prime}\left(b+b^{\prime}\right)$
- =a'b'c'd'+a'b'c'd+a'b'cd'+a'b'cd+a'bcd+a'bcd+ abcd'+ab'cd'
- Note: a minterm expression is not necessary the simplest expression.


## Minterm and Maxterm Expansions (9/10)

- Converting a general logic expression into a maxterm expression
- Through repeatedly applying $X X^{\prime}=0$


## Minterm and Maxterm Expansions (10/10)

- When comparing two logic expressions, you can convert both into their minterm expressions and then compare.
- Example:
$-a^{\prime} c+b^{\prime} c^{\prime}+a b$ and $a^{\prime} b^{\prime}+b c+a c^{\prime}$


## General Minterm and Maxterm Expansion (1/3)

- A minterm expansion for an $n$ variable function can be represented as a $2^{n}$ long vector
- Example:
$-\mathrm{F}(\mathrm{A}, \mathrm{B}, \mathrm{C})=a_{0} m_{0}+a_{1} m_{1} \ldots a_{6} m_{6}+a_{7} m_{7}=\sum_{i=0}^{7} a_{i} m_{i}$
- If $a_{i}=1, m_{i}$ is present in the expression.


## General Minterm and Maxterm Expansion (2/3)

- Similarly, a maxterm expansion for an n variable function can also be represented as a $2^{n}$ long vector
$-\mathrm{F}(\mathrm{A}, \mathrm{B}, \mathrm{C})=\left(a_{0}+M_{0}\right)\left(a_{1}+M_{1}\right) \ldots\left(a_{6}+M_{6}\right)\left(a_{7}+M_{7}\right)=\prod_{i=0}^{7}\left(a_{i}+M_{i}\right)$
- If $a_{i}$ is $0, M_{i}$ is present in the expression. Why?


## General Minterm and Maxterm Expansion (3/3)

- Given two different minterm expansions of $n$ variables

$$
\begin{aligned}
& f_{1}=\sum_{i=0}^{2^{n}-1} a_{i} m_{i} \\
& f_{2}=\sum_{j=0}^{2^{n}-1} b_{j} m_{j} \\
& f_{1} f_{2}=\left(\sum_{i=0}^{2^{n}-1} a_{i} m_{i}\right)\left(\sum_{j=0}^{2^{n}-1} b_{j} m_{j}\right)=\sum_{i=0}^{2^{n}-12^{n}-1} \sum_{j=0} a_{i} b_{j} m_{i} m_{j}=\sum_{i=0}^{2^{n}-1} a_{i} b_{i} m_{i}
\end{aligned}
$$

## Incompletely Specified Functions (1/5)

- Sometimes, not all the combinations of the inputs are considered in the circuit.
- Unconsidered combinations are referred to as "do not care" terms.
- In the truth table, the outputs for "do not care" terms are designated ' X '


## Incompletely Specified Functions (2/5)

- We could ignore the "do not care" terms, then the logic expression is
- $F=A^{\prime} B^{\prime} C^{\prime}+A^{\prime} B C+A B C$
- =A'B'C' ${ }^{\prime}$ BC

| $A$ | $B$ | $C$ | $F$ |
| :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | $X$ |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | $X$ |
| 1 | 1 | 1 | 1 |

## Incompletely Specified Functions (3/5)

- It does not matter, if we assign 1/0 to Xs
$-F=A^{\prime} B^{\prime} C^{\prime}+B C+A^{\prime} B^{\prime} C$
$-=A^{\prime} B^{\prime}+B C$ Simpler expression

| $A$ | $B$ | $C$ | $F$ |
| :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## Incompletely Specified Functions (4/5)

- Sometimes, assigning 1 to X's may contribute to simplifying the logic expression.


## Incompletely Specified Functions (5/5)

- In a minterm expansion, the "do not care" terms are denoted $\boldsymbol{d}$

$$
F=\sum m(0,3,7)+\sum d(1,6)
$$

- In a maxterm expansion, the "do not care" terms are denoted $D$

$$
F=\prod M(2,4,5)+\prod D(1,6)
$$

## Design of Binary Adders and Subtracters (1/7)

- Full Adders



## Design of Binary Adders and Subtracters (2/7)

- Sum=m1+m2+m4+m7
- Sum $=X^{\prime} Y^{\prime} C_{i n}+X^{\prime} Y C_{\text {in }}{ }^{\prime}+X Y^{\prime} C_{\text {in }}{ }^{\prime}+X Y C_{\text {in }}$
$-=X^{\prime}\left(Y^{\prime} C_{i n}+Y C_{\text {in }}{ }^{\prime}\right)+X\left(Y^{\prime} C_{\text {in }}{ }^{\prime}+Y C_{\text {in }}\right)$
$-=X^{\prime}\left(Y\right.$ xor $\left.C_{\text {in }}\right)+X\left(Y \text { xor } C_{\text {in }}\right)^{\prime}$
- $=X \operatorname{xor}\left(Y\right.$ xor $\left.C_{\text {in }}\right)=X$ xor $Y$ xor $C_{\text {in }}$
- $\mathrm{C}_{\text {out }}=\mathrm{m} 3+\mathrm{m} 5+\mathrm{m} 6+\mathrm{m} 7$

$$
\begin{aligned}
& -C_{\text {out }}=X^{\prime} Y C_{\text {in }}+X Y^{\prime} C_{\text {in }}+X Y C_{\text {in }}^{\prime}+X Y C_{\text {in }} \\
& -=Y C_{\text {in }}+X C_{\text {in }}+X Y Y
\end{aligned}
$$

## Design of Binary Adders and Subtracters (3/7)

Sum $=\mathrm{X}$ xor Y xor $\mathrm{C}_{\text {in }}$
Full Adder

$$
C_{\text {out }}=Y C_{\text {in }}+X C_{\text {in }}+X Y
$$



## Design of Binary Adders and Subtracters (4/7)

- Four full adders can be used to make a 4 bit binary adder



## Design of Binary Adders and Subtracters (5/7)

- When adding two signed number, overflow must be considered.
- Adding two positive numbers gives a negative number: $\mathrm{A}_{3} \mathrm{~B}_{3} \mathrm{~S}_{3}{ }^{\text {, }}$
- Adding two negative numbers gives a positive number: $A_{3}{ }^{\prime} B_{3}{ }^{\prime} S_{3}$
$-V=A_{3}{ }^{\prime} B_{3}{ }^{\prime} S_{3}+A_{3} B_{3} S_{3}$ ' can be used to reflect if overflow occurs


## Design of Binary Adders and Subtracters (6/7)

- Binary subtracter using full adders
- Remember two's complement
- Reverse all the bits
- Add 1


## Design of Binary Adders and Subtracters (7/7)



