CL=CSCI 160

CLASS 10

Review **Topics**

(1) Number Representations

- (r - 1)'s, r's complements, algorithm for subtraction, etc
- Find basis x if $(. . .)_x = . . .$

(2) (3) – Boolean algebra:

- Expression manipulation
- Boolean operators

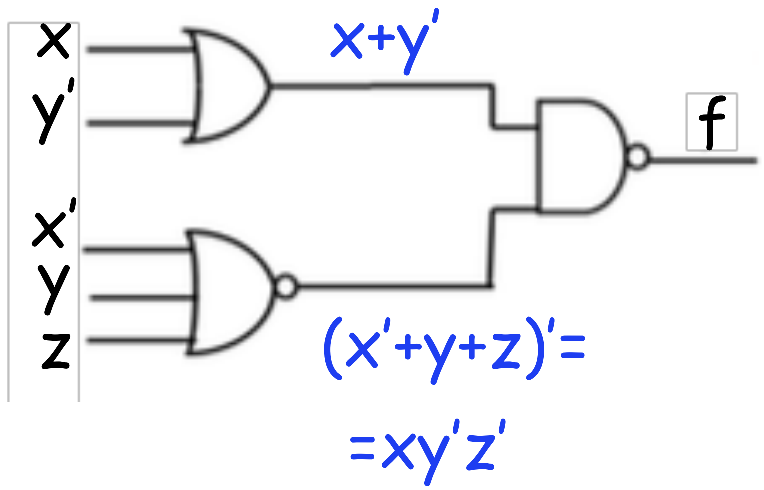
(4) distribution (both)
(12) absorption
(10) De Morgan

(4) – Circuit – logic diagram (gates)

B. functions; truth table

(4) Gates

Find the truth table + expression of f



$$f = ((x+y')xy'z')' =$$

$$= (x+y')' + x' + y + z =$$

$$= \underbrace{x'y + x'}_{x'\ (12)} + y + z = \boxed{x' + y + z}$$

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

What can we say about the truth table for f?

When do we have f = 0, and when is f = 1?

It is 0 exactly when x'=0, and y=0, and z=0, that is

for the triple x y z = 1 0 0.

In rest we have f = 1.

(3) Operators

HW 8.1

(a) Prove: Inhibition is not associative

We chose $F_2 = x/y = x\,y'$

F_2 associative $\overset{\text{Det}}{\Leftrightarrow}$ $F_2(F_2(x, y), z) = F_2(x, F_2(y, z))$ $\forall$ **x, y, z**

$$\Updownarrow$$

$$F_2(xy', z) = F_2(x, yz')$$

$$\Updownarrow$$

$$xy'z' = x(yz')'$$

$$\Updownarrow \text{(10)}$$

$$xy'z' = x(y'+z) \qquad \forall\, x, y, z$$

F_2 not associative $\overset{\text{Det}}{\Leftrightarrow}$ $\exists$ **x, y, z:** $xy'z' \neq x(y'+z)$

Make, say, RS = 1, while LS = 0: We need: $x = 1$ and ($y' = 1$ or $z = 1$)

There are multiple such counterexamples.

We only need one, for instance $x = 1$, $z = 1$, and any value for y, say $y = 0$.

We have LS = $0 \neq 1$ = RS. We just proved: $\boxed{\textbf{F}_2\textbf{ is not associative}}$

(3) Operators - continued **HW 8.1 - continued**

(b) Prove: Exclusive OR (XOR) is commutative

$F_6(x, y) = x\,y' + x'\,y = x \oplus y$

F_6 commutative $\overset{\text{Det}}{\Leftrightarrow}$ $F_6(x, y) = F_6(y, x) \quad \forall\, x, y$

$\Updownarrow$

$\oplus$ commutative $\overset{\text{Det}}{\Leftrightarrow}$ $x \oplus y = y \oplus x \quad \forall\, x, y$

$\Updownarrow$

$x\,y' + x'\,y = y\,x' + y'\,x \quad \forall\, x, y$

The equality holds for both binary operations + and • are commutative.

We just proved: $\boxed{\text{F}_6\text{, that is Exclusive OR (XOR) is commutative}}$

(2) Expression manipulation

Basically what we have been doing all along - old HW

2.6)

a) $(xy' + x'y)' \overset{(10)}{=} (x' + y) \cdot (x + y') = \boxed{xy + x'y'}$

b) $((AB' + C)\,D' + E)' \overset{(10)}{=} ((A' + B) \cdot C' + D)\,E' \overset{(4)}{=} \boxed{(A' + B + D)(C' + D)\,E'}$

c) $((x + y' + z)(x' + z')(x + y))' \overset{(10)}{=} \boxed{x' \cdot y \cdot z' + x \cdot z + x' \cdot y'}$

More exercises:

$x'y$

A. $xyz' + x'yz + xyz + x'yz' \overset{(7)}{\underset{(4),(5),(2)}{=}} xy + yz + x'y \overset{(4)}{\underset{(5),(2)}{=}} y + yz \overset{(12)}{=} \boxed{y}$

y

$xy\underbrace{(z+z')}_{1} = xy$

$(x+x')yz$

B. $\underbrace{xyzt + x'yt + xyz't}_{xyt} + yzt \underset{(5)\,(2)}{\overset{(4)}{=}} \underbrace{xyt + x'yt}_{yt} + yzt \underset{(5)\,(2)}{\overset{(4)}{=}} yt + yzt \overset{(12)}{=} \boxed{yt}$

C. Prove: $xy + x'z + yz = xy + x'z$

Expand the terms that are in LS \ RS, namely yz

$\boxed{LS} = xy + x'z + (x + x')yz = xy + x'z + xyz + x'yz \overset{(12)\times 2}{=}$

$= xy + x'z = \boxed{RS}$

(1) Number Representations

Review conversions and operations in any base.

Let's do the r's / (r-1)'s complements and apply the algorithm for subtracting two numbers.

1. r = 16

Compute the (r-1)'s complement of $3ABF09_{hex}$. We have r - 1 = 16 - 1 = 15. Subtract each digit from F=15:

$$(3ABF09_{hex})_{15'C} = \boxed{C\,5\,4\,0\,F\,6}$$

Let's do it via r = 2 (all computations are in binary). How?

$$3ABF09_{hex} = 0011\ 1010\ 1011\ 1111\ 0000\ 1001_2 \xrightarrow{\text{1's complement}}$$

$$\xrightarrow{\text{1's complement}} 1100\ 0101\ 0100\ 0000\ 1111\ 0110_2 = \boxed{C540F6_{hex}}$$

2. Let's apply the algorithm for subtracting two numbers

$r = 8$ $\quad n = 4,\ m = 2$ $\quad$ Remember: $N = \overbrace{\qquad\qquad}^{\text{integer part}} \bullet \overbrace{\qquad\qquad}^{\text{fraction part}}$

Alg (M-N): 1) $M + N_{(r-1)c}$ $\quad$ 2) a) e.a.c. -> add it to l.s.d.
b) no e.a.c. -> compute -(r-1)'s compl. of 1)

Let $M_8 = 1671.02$, $N_8 = 2016.12$ Perform M – N using the algorithm using (r-1)'s rep.

1st solution-direct:

```
         1 1
   M8 =  1671.02   +
  N7c =  5761.65
  ----------------
         7652.67
```

No e.a.c., branch 2b), so do – 7'c

Result: $\boxed{-0125.10_8}$

2nd solution-convert to binary:

M_2: 001 110 111 001. 000 010
N_2: 010 000 001 110. 001 010

```
         1 1  1 1 1   1      1
  M2:    001 110 111 001. 000 010  +
  N1C:   101 111 110 001. 110 101
  ---------------------------------
         111 110 101 010. 110 111
```

No e.a.c., branch 2b), so do – 1'c

–000 001 010 101.001 000 Convert to base 8 to get same result as above.

NOTE

We can do these conversions to other bases and back only if self-complementary:

hex ⟷ binary Yes, for self-complementary

Decimal ⟷ (2421) code
Decimal ⟷ (8, 4, -2, -1) code ⟷ (2421) code
(8, 4, -2, -1) code Yes, for self-complementary

Decimal ~~⟷~~ BCD No, not self-complementary

see next slide!

Decimal $\longleftrightarrow$ (2421) code

(8, 4, -2, -1) code   Yes, for self-complementary

Remember:

Decimal $\nleftrightarrow$ BCD   No, not self-complementary

Table

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combinations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110