Lab 1

decimal to binary

a) 0000 - 0011 3

b) 0000 - 1001 9

c) 0000 - 1111 15

d) 0001 - 0010 18

e) 0000 - 0111 7

f) 0000 - 1010 16

g) 0000 - 0101 5

h) 0001 - 0000 16

i) 0001 - 0110 22

j) 0001 - 1101 29

k) 0010 - 0011 37

l) 0100 - 0000 64

Binary to decimal

A) 0010 0101 → 37
(32, 4, 1)

B) 0100 1111 → 79
(64, 8, 4, 2, 1)

C) 0001 0110 → 22
(16, 4, 2)

D) 0001 0010 → 18
(16, 2)

E) 0001 1011 → 27
(16, 8, 2, 1)

F) 0000 1100 → 12
(8, 4)

G) 0011 0001 → 49
(32, 16, 1)

H) 0010 1111 → 47
(32, 8, 4, 2, 1)

H) 0010 1111 → 47
(32, 8, 4, 2, 1)

I) 0100 1000 → 72
(64, 8)

J) 0110 0001 → 97
(64, 32, 1)

K) 1010 0110 → 166
(128, 32, 4, 2)

L) 1011 0010 → 178
(128, 32, 16, 2)

convert to Hex

A) 0110_0107

Hex = C5

B) 01001111 → 4F

C) 00010110 → 16

D) 00012 0010 → 12

E) 1000_1011 → 8B

F) 00111100 → 3C

G) 0011_0010 → 32

H) 0016 0201 → 25

I) 0 1810 1010 → 4A

J) 01101011 → 6B

K) 10101110 → A6

L) 10111011 → BB

Hex → Binary

A) 0015

0000_0000_0001_0101

B) 001F

0000 0000 0001 1111

C) 0111

0000 0001 0001 0001

D) 0018

0000 0000 0001 1000

E) 000D

0000 0000 0000 1101

F) 040C

0000 0100 0000 1100

G) 0A30

0000 1010 0011 0000

H) 06EAh

0000 0110 1110 1010

I) 1CA3

0001 1100 1010 0011

J) 0B33h

0000 1011 0011 0011

K) 03B1

0000 0011 1011 0001

L) 2DBA

0010 1101 1011 1010

Hex → Dec

A) 00C5

0 + 0 + 12·16 + 5·1

192 + 5 = 197

B) 001F

0001 1111

16 + 15 = 31

C) 0111

0000 0001 0001 0001

256 16 1

273

D) 0213

0000 0010 0001 0011

512 16 2 1

E) 063C

0000 0000 1100 1100

32 16 8 4 → 60

F) 040C

0000 0100 0000 1100

1024 → 1036

8 4

G) 0A30

0000 1010 0011 0000

2048 512 32 16

2560

32

2592

40 ✓

2600

H) 06EA

0000 0110 1110 1010

1024 512 128 64 32 8 2

1536

128

1664

64

1728

32

1760 → 1770 ✓

I) 1CA3

0001 1100 1010 0011

4096 2048 1024 128 32 2 1

7331

J) 0B33h

0000 1011 0011 0011

2048 512 256 32 16 2 1

2867

K) 03B1

0000 0011 1011 0001

512 32 16 1

128

L) 2DBA

0010 1101 1011 1010

8192 2048 1024 256 128 8 2

11706 ✓

4- ADDING BINARY

(A) 128 64 32 16 8 4 2 1
1 0 1 0 1 1 1 1 (175)
+ 1 1 0 0 0 0 1 1 (195)
1 0 1 1 1 0 1 0 = 370
256 64 32 16 2

(B) (57)
0011_1001
1111_0011 (243)
1 0 0 1 0 1 1 0 0 (300)
256 32 8 4

(C) (255)
1111_1111
1110_1110
1 1 1 0 1 1 0 0 (492)
256, 128, 64, 32, 8, 4

#7 Signed 8 bit integers to decimal

(A) 1111_1111
Negative
1's compliment
1111_1111
0000_0000
add 1
0000_0001
= -1 in decimal

(B) 1111_0000 neg
0000_1111
+1
0001_0000
= -16 in decimal

(C) 1000_0000 neg
0111_1111
+1
1000_0000
-128

(D) 1000_0001 neg
0111_1110
+1
0111_1111
64 32 16 8 4 2 1
= -127
positive

(G) 0000_1111
8 4 2 1
= 15

(H) 1010_1111 neg
0101_0000
+1
0101_0001
64 16 1
-81

(I) 1111_1100 neg
0000_0011
+1
0000_0100
= -4

(J) pos 64 16 4 1
0 1 0 1 0 1 0 1
= 85

#8 write each of the following signed decimal integers in 8-bit binary notation.

(A) -2
(abs = 2)
0000_0010
1111_1101
+1
1111_1110 ✓

(B) -7 (abs)
0000_0111
1's comp 1111_1000
+1
1111_1001

(C) -128
abs
1000_0000
0111_1111
+1
1000 0000

(D) -16 abs
0001_0000
1110_1111
+1
1111 0000

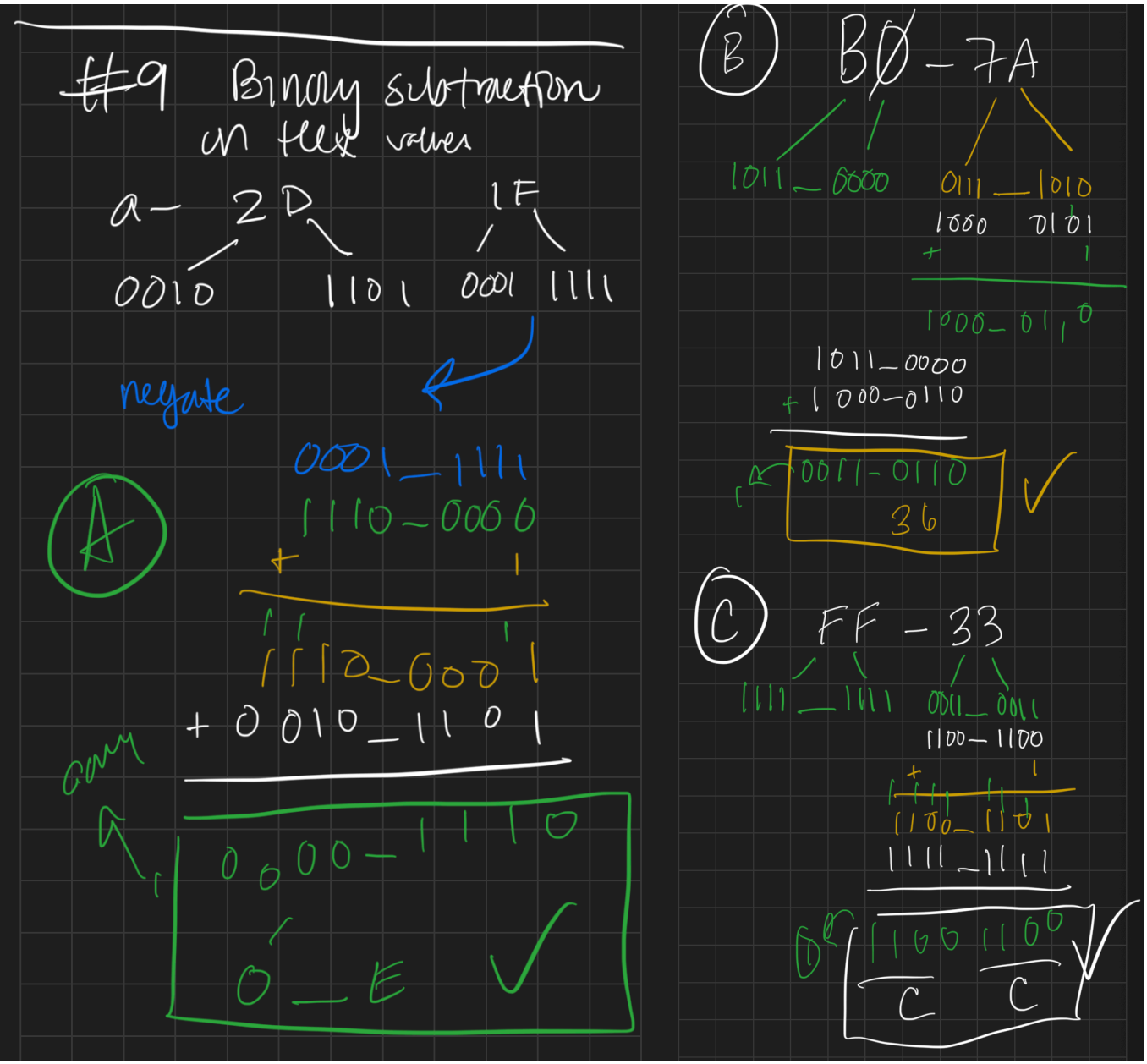
(E) 15
0000_1111

(F) -1 abs
0000_0001
1111_1110
+1
1111_1111

(G) -56 (abs)
0011_1000
1100_0111
+1
1100_1000

(H) 127
0111_1111

#9 Binary subtraction on hex values



#10

(52) + (65) = 117 ✓

34 + 41

0011 0100 0100 0001

```
  0100_0001
+ 0011_0100
-----------
  0111_0101
```

(A)

64 + 32 + 16 + 4 + 1

96
16
112 + 5 = 117 ✓

(B)

18 + 42 → 66 = 90

0001 1000 0100 0010

16 8 64 2

```
  0001_1000
+ 0100_0010
-----------
  0101_1010
```

64 16 8 2 = 90 ✓

(C)

AC + CC

1010_1100 1100_1100

128 32 8 4 → 172

128 64 8 4 → 192, 12 → 204

204
172
Actual → 376 → 8 bit though

```
  1010_1100
+ 1100_1100
-----------
  0111_1000
```

overflow

64 32 16 8 → 96 24

Dec → 120

Hex → 78

OOPS need to use Base 16 method

```
  34
+ 41
----
  75
```

0111 0101

64 32 16 → 48; 4 1 → 5

53

117 ✓

(B)

```
  18
  42
----
  5A
```

0101_1010

64 16 8 2

10, 26, 90 ✓

```
  AC   -12
+ CC   -12
--------
  178
```

24/16 = 1r8

23/16 = 1r7

0001_0111_1000

(D)

```
  2B
+ A7
----
  D2
```

18/16 = 1r2

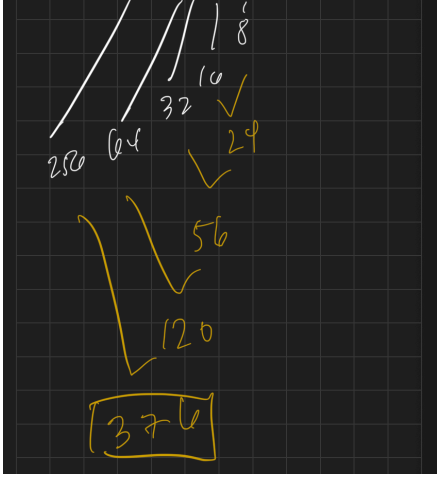
1101_0010

128 64 16 2

18

82

210



#11

A) 0x1234

next byte

→ 0x1235

B) 0xCAFE

with 16 bit word (2 bytes)

CAFE 16/16

\+ 2 1 r 0

CB00

D) 400

0100 0000 0000

10 9 8 7 6 5 4 3 2 1 0

1024

400

− 492

32

3E0

0011 1110 0000

512 256 32 64 128

768

192

224

768

492

3E0

− 320

0010 0000

C) 0XDEAD

32 bit = 4 bytes

DEAD 17/16 r1

\+ 4

DEB1

\+ 4

DEB1

E) 0x100

64 bits = 8 bytes

= 8?

@ 108

13 — 000000

1 2 3 4 5 6

16 8 4 2 1

32

31 < 40

Thus 6 digits is min

12 Design your own numbering system [Base 1]

A) Conversion to decimal

- start at base $10^{n=0}$
- increment n by 1 for every tally gone.
- when get to end of current base n - increase power.
- shift left and repeat until tallys gone.

B) Conversion from decimal

123

$1 \cdot 10^2$ tally marks $2 \cdot 10^1$ tally marks $3 \cdot 10^0$ tally marks

Adding 2 numbers

one must simply gather all the tally marks into one location

+ =

c) conversion to binary.

Ex1

1010

- all pairs balanced so least sig = 0
- then find greatest power of 2 in the set
- make that position as 1 in binary
- tally the remaining and work that out in binary

Ex2

1100

- all pairs matched
- 0
- largest power of 2 present

1010

conversion from binary

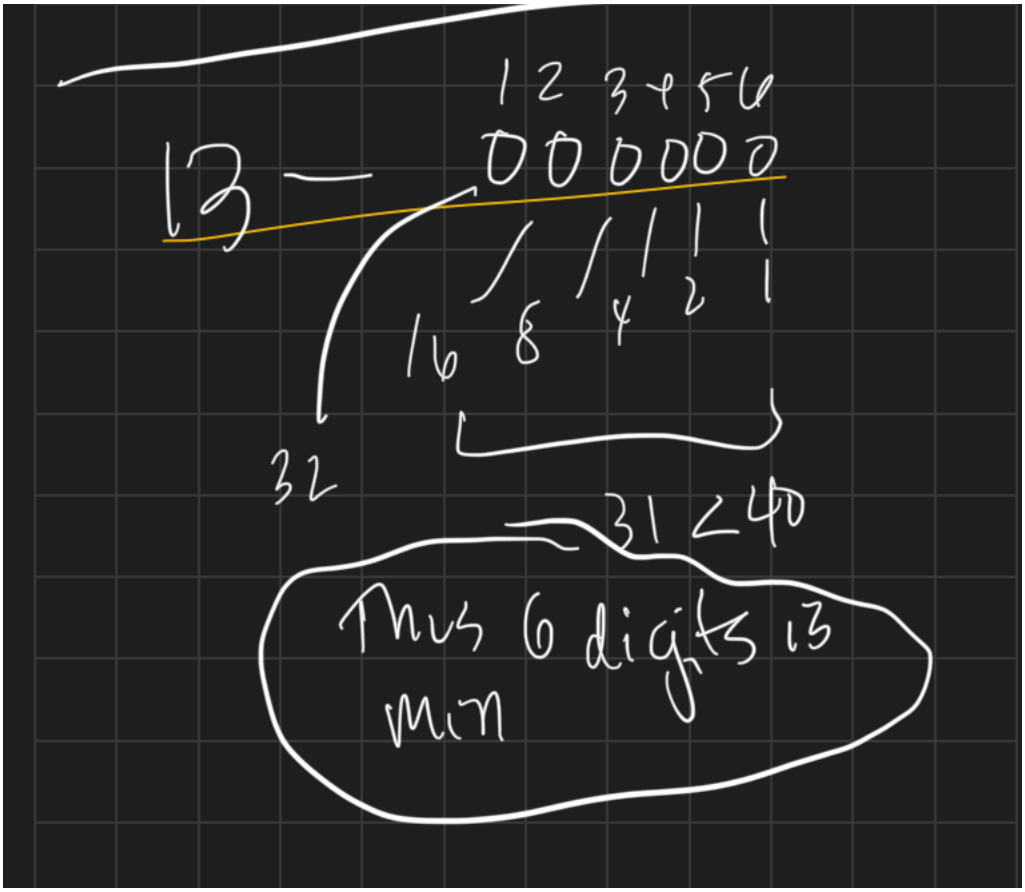
1010

no tallys

$0 \times (1)^0$

$1 \times (0)^2$ $1 \times (1)^1$

$1 \times (1)^3$ no tallys


1 2 3 4 5 6
13 — 0 0 0 0 0 0
16 8 4 2 1
32
31 < 40
Thus 6 digits is min