b) fractional 0.4 mantissa

\[
\begin{align*}
00001100. & \Rightarrow 0.0110 \times 2^5 \\
12 & \Rightarrow 0.375 \times 32
\end{align*}
\]

\[
\text{Ex: a) } 11010000 \Rightarrow 1010. \times 2^{11} = 3
\]

\[11010000 \Rightarrow 101.0 \times 2^{11} = 32\]

Case 2

Special cases

1) All zeros 00000000

Mantissa position does not matter

How most regular & simpler

2) All ones 1111111[1000]

a) choose mant. after "removing" max sign bits

\[
1000. \times 2^{-3} = -8 \times \left(\frac{1}{8}\right) = 1\]

\[\checkmark\]
5) Choose mant. - remove max sign bits
   - preserve bits

```
  1 1 1 1 1
```

1 1 1 1 0 \times 2^0 = -1 \times 1 = [1-1] \checkmark

+ Choose regular HW

```
Adders + subtractors
```

```
<table>
<thead>
<tr>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
</tr>
<tr>
<td>m</td>
</tr>
</tbody>
</table>
```

```
Adders
```

width of sum = width of input + 1

- 2\^ input
- both input equal width

A) Carry-Propagate (CPA)
- "normal" addition
- output single word
- carry ripples across word

```
+ 
```
1) **Ripple-Carry Adder (RCA)**
   - simplest
   - smallest
   - slowest

2) **Faster CAs**
   - Carry Select
   - Carry Look-ahead

---

**Subtraction**

- Requires a signed format
  - $A - B = A + (-B) = A + (\neg B) + 1$

---

Add/Sub

```
A
+     +
|     |
B     +
|     |
+     |
|     |
A + B
```

---

```
add/sub
```

---

```
A
+     +
|     |
B     +
|     |
+     |
|     |
A+B
```
Efficient Multi-Input Addition

1) Alg., format conv., alignment, sign ext., rounding...

2) dot diagram

3) cover w/ carry-save adders

4) repeat until 2 output terms

5) CPA

6) Reminders
   - output width
   - sign exten.
   - only carry, not output bits

Generally better to commit data earlier than later
0.4.

```
00111
+---+
3   |
01 1
10 1
```

```
\begin{align*}
\{01 & 0 \\
\{10 & 0 \\
\end{align*}
```

```
M \quad \text{unit}
```

```
\begin{array}{c}
\quad\
\quad\
\end{array}
```