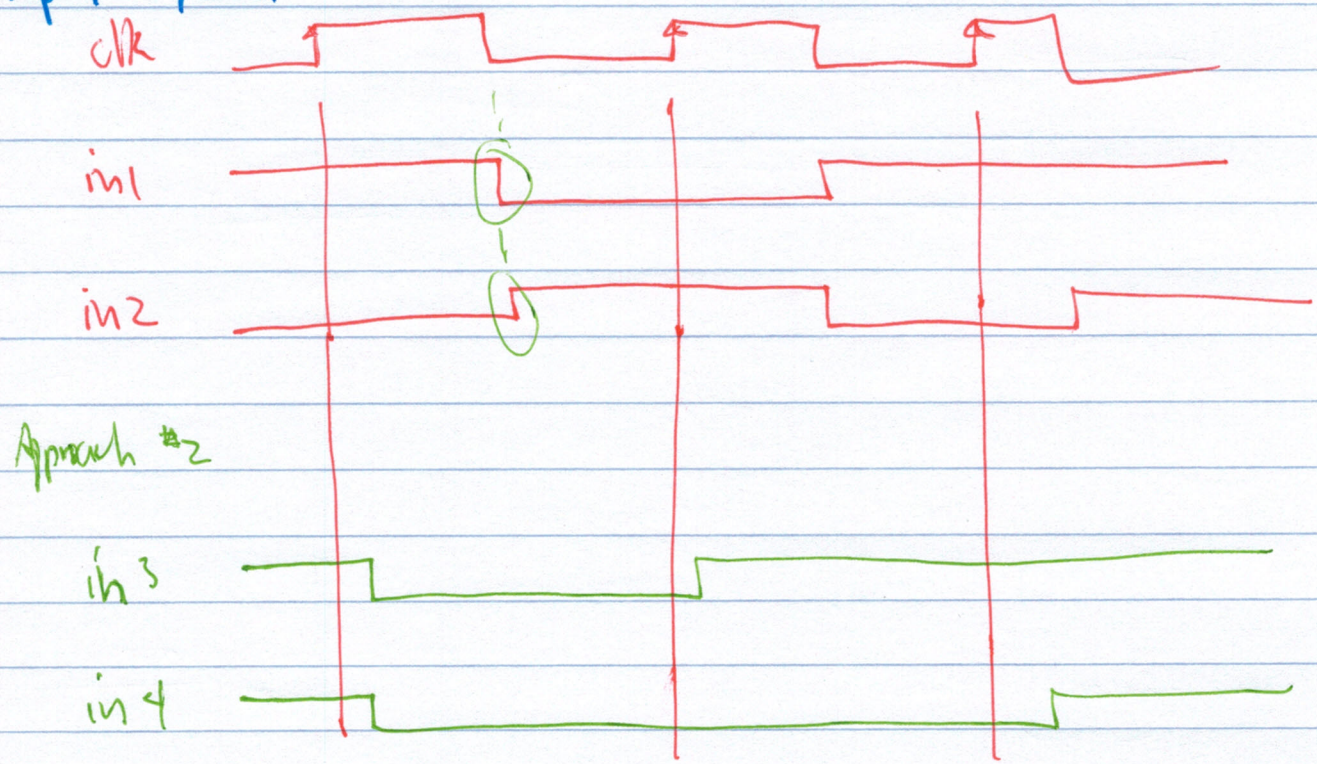
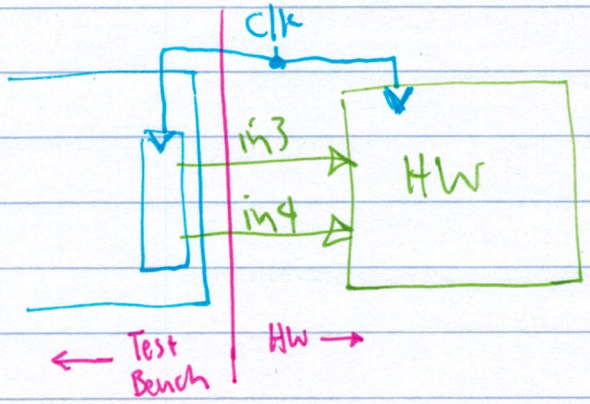
EEC 180

April 15, 2021





3 Reasons for Test/Verification

* ① Design time: is design correct?
  - designer time is critical

② Manufacturing time

Pass / Fail

Ship / Scrap or Repair

- speed is critical
- coverage is critical

③ Diagnostic
- What exactly is failing?
- Pinpoint the fault

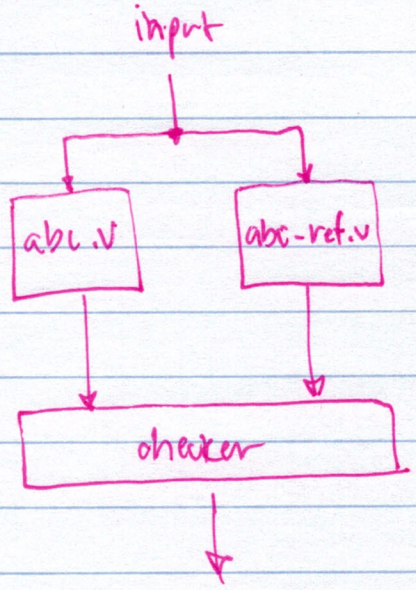
How to verify designs?

1) Eyeball text printouts
   - Quickest and easiest

2) Eyeball waveforms

3) "Golden Reference" approach

   Ref. written in a different way



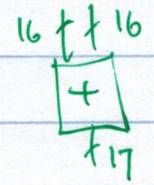
Checker

1) "Bit accurate"
   + easy to automate
   + less testing than #2

2) "close enough"
   + simpler golden ref.
   - complex comparison

Generating Test Cases

1) Exhaustive - test all input cases



Ex: 16-bit adder - 32 inputs

2^{32} (4.3 billion) possible inputs

1 million tests/sec → 71 minutes

Ex: 32-bit adder

1 million tests/sec → 584,942 years

2) Directed - chosen by hand

Ex: adder 8-bit

0+0, 0+1, 1+0, 0+(-1), (-1)+0, ...

(-1) + (-1)

127 + 127

(-128) + (-128)

3) Random

- as much HW (computing)
- as long as possible
- think about debugging

Binary Numbers

1) Unsigned n-bit range $[0, 2^n - 1]$

$$\text{value} = \sum_{i=0}^{n-1} a_i b^i \quad , b = 2$$

X	X	X	X
2^3	2^2	2^1	2^0
8	4	2	1

2) Sign magnitude

- two zeros: +0, −0
- S=1 → neg. S=0 → pos.

3) 2's complement

X	X	X	X
−8	4	2	1

- negative MSB positional weight
- range $[-2^{(n-1)}, +2^{(n-1)}-1]$

$[-8, +7]$, $n=4$

4) 1's complement

5) Binary-Coded Decimal (BCD)

- Each base-10 digit is rep. by 4 bits
- 7-seg displays
- keypads

Ex: 0000_0101

=	$= 5_{10}$	unsigned
+	$= +5_{10}$	sign mag
	$= +5_{10}$	2's compl.
	$= 05$	BCD

Ex: 1000_0011 $= 131_{10}$ unsigned

$= -3$ sign mag.

$= -125$ 2's compl.

$= 83$ BCD

A) Integer

B) Fractional

- f = # of fractional bits value = $\sum_{i=0}^{n-1} a_i b^{i-f}$
- unsigned, 2's compl., sign mag.
- Ex: 10100.001 "5 dot 3 format"

$2^4\ 2^3\ 2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}\ 2^{-3}$ unsigned: $20\frac{1}{8}$

16 8 4 2 1 . 1/2 1/4 1/8

"no decimal print in HW"

C) Full fractional

Ex: 0.4

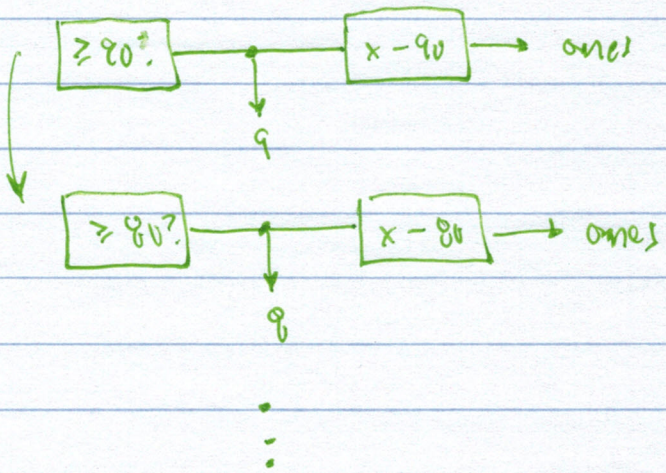
Convert BCD → unsigned

Ex: 135_{BCD}

1×100	0110_0100
3×10	0001_1110
9×1	+ 0000_0101
	1000_0111 = 135 ✓

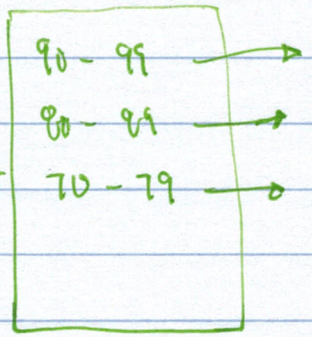
Unsigned → BCD

Ex: input < 100



Ex: 93

unsigned



9 x-90 = 3