# EEC 180



June 1, 2021

## System Design

① Specification

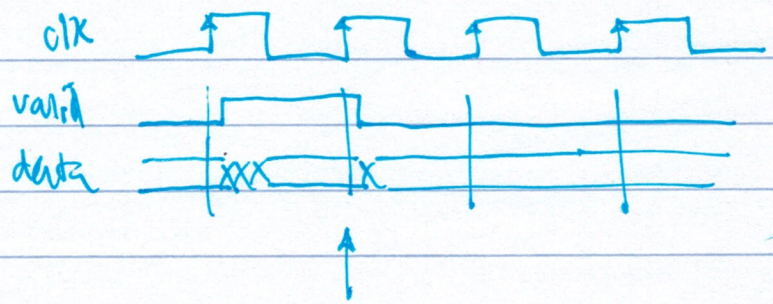
a) what the system is
" " " does
how it is used

b) inputs / outputs
- formats of numbers : unsigned, 2's complement, 6.2 format, BC etc.
- range of values
  Ex: -1, +1
  → choose 2's compl 2-bit →

| | |
|---|---|
| 01 | +1 |
| 00 | 0 |
| 11 | -1 |
| 10 | -2 |

- timing

Ex:




- protocols
  Ex: parity

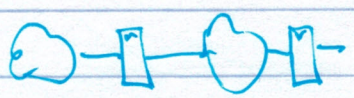
c) All user visible state
- configuration registers
- mode bits
- internal memories

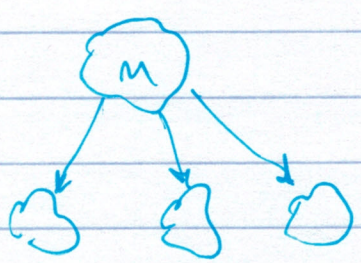
d) All modes of operation
- test
- safe
- normal
- advanced features

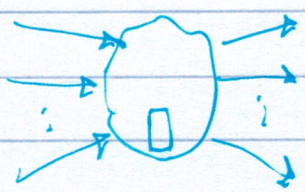
e) Notable features of the system
- MP3
- HDMI
- bluetooth

f) All interesting "edge cases"
- Ex: multiple switches active
- Ex: sensor failure

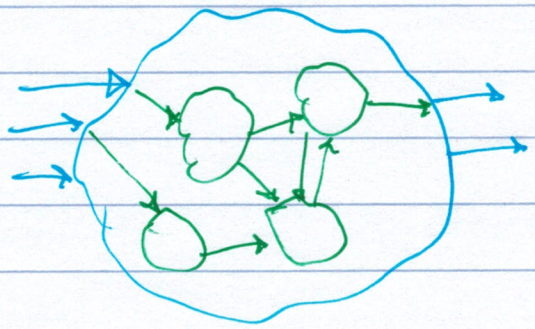
② Partitioning
- state
- task → • pipelining
  • master/slave partitioning
  • resource partitioning
- interface



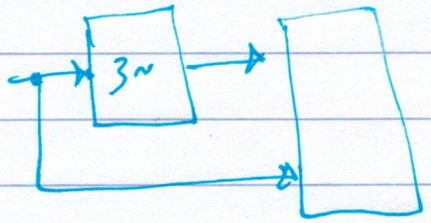


③ Interface Specification
(internal modules)



④ Timing design

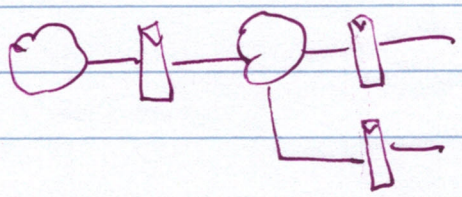



⑤ Module design

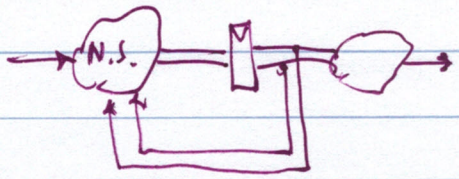
⑤.5 Debugging

⑥ Performance Tuning

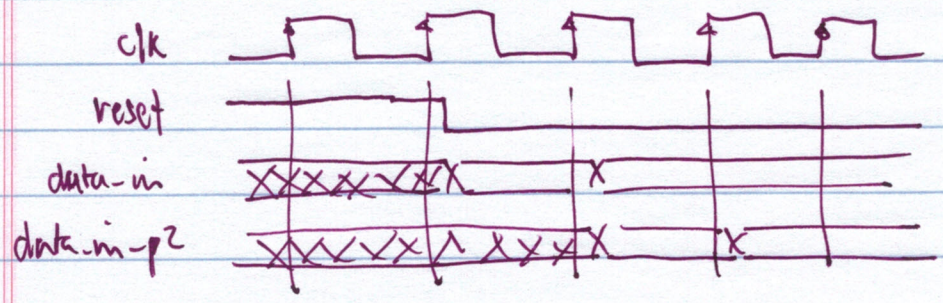
① Design a pipelined block diagram

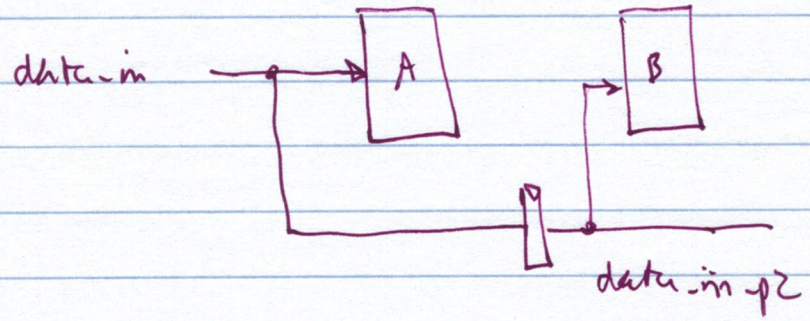


Controller circuitry
- counters
- FSMs



② Timing diagrams
- All system inputs
- All key internal signals
- All system outputs



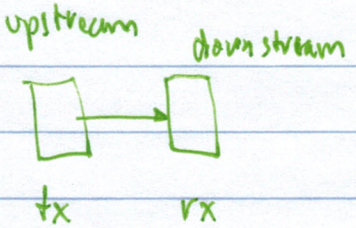

③ Iterate on #1 and #2

④ Design controllers
- Counters + FSMs
- State Graphs

⑤ Stare at diagrams until you are sure of your design

⑥ Begin thinking about verilog.
Type it in. Debug.

Interface timing and protocols



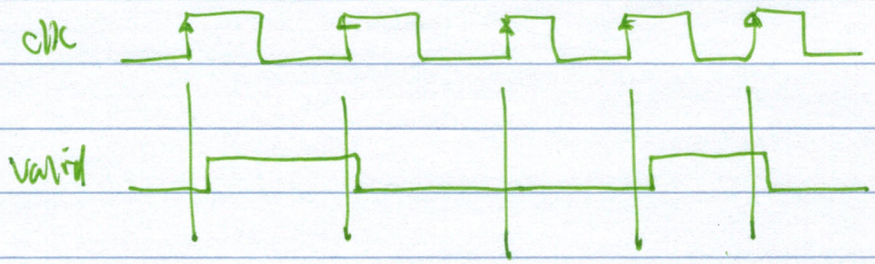

① Always valid

Ex: temp. sensor

Samples can be dropped or duplicated

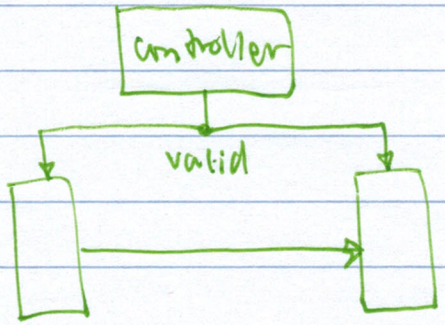
② Periodically valid - every X cycles $X \geq 2$

Ex: $X = 3$

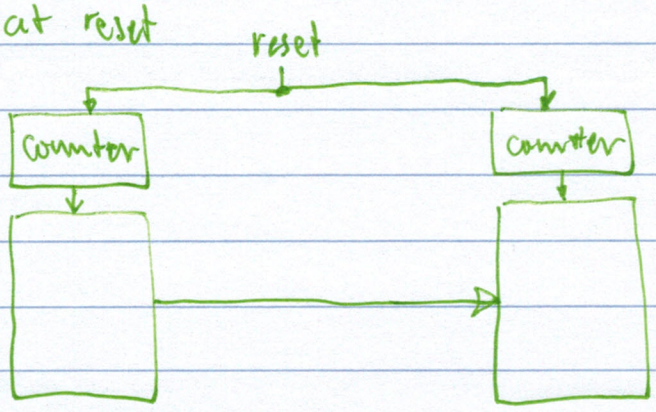



sender + receiver must synchronize

a)



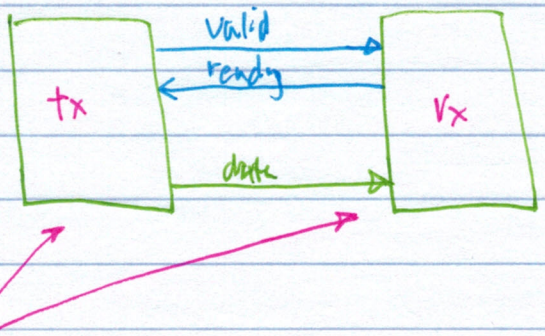

b) synch at reset



## ③ Flow control (dynamic)

flexible but:

- data timing is not known (e.g. user input)
- " " " data dependant (e.g. video compression)
- " " " too complicated to model efficiently




Both sides know what is going on

| valid | ready | |
|---|---|---|
| 0 | 0 | idle |
| 0 | 1 | rx ready |
| 1 | 0 | tx ready |
| 1 | 1 | data transferred |

a) "Pull Timing"

Transmitter always ready

RX controls data flow

b) "Push Timing"

RX always ready

TX controls the data flow