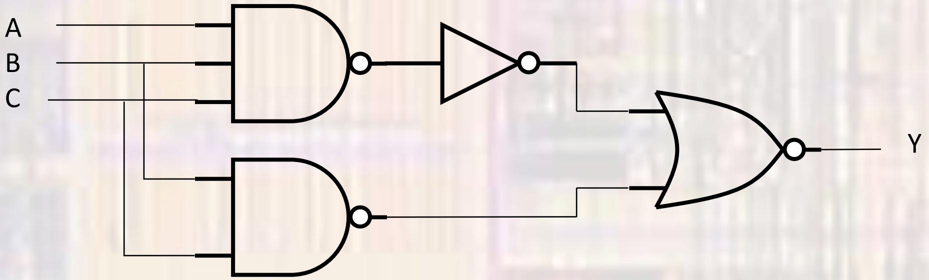
Register Transfer Architecture

Last updated 1/18/21

Register Transfer Architecture

- Combinational Logic
  - Outputs are dependent only on current inputs
  - Output changes can be triggered by any input change



Register Transfer Architecture

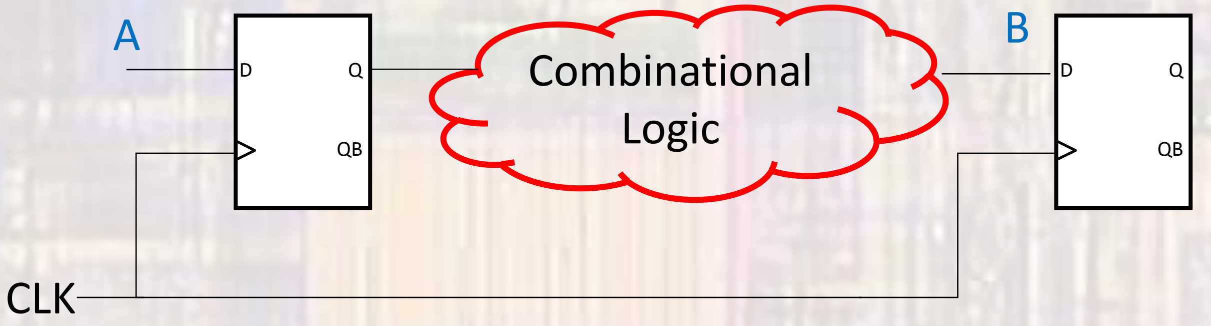
- Sequential Logic
  - Outputs are dependent on inputs and current STATE
    - State
      - Collection of outputs and intermediate values stored within the system
      - Requires some sort of memory
  - Asynchronous - Output changes can be triggered by some input changes
  - Synchronous – Outputs changes triggered by a synchronous event – usually a clock

Register Transfer Architecture

- State memory elements
  - Bi-stable circuit
  - Latches
  - Flip-flops
  - Registers
  - Memories
    - ROM
    - RAM
    - SRAM
    - SDRAM
    - Flash

Register Transfer Architecture

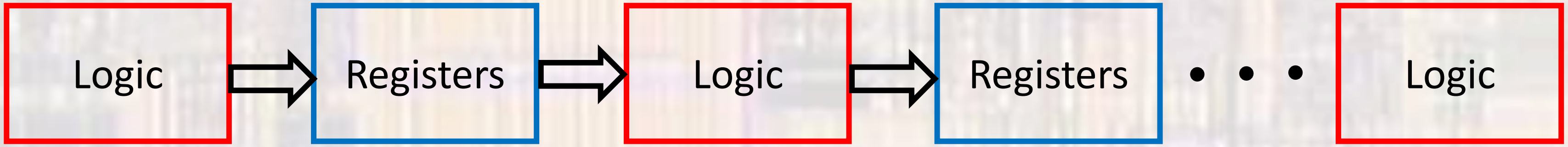
- Typical Flip-Flop Circuit



- Need the data to get from point A to point B before the next clock edge occurs

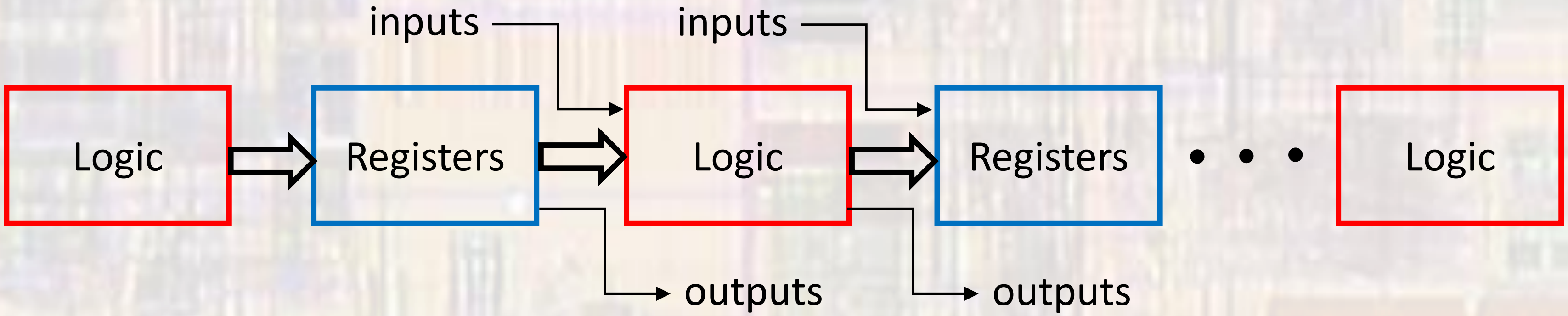
Register Transfer Architecture

- Register Transfer Architecture
  - Synchronous System
  - Blocks of combinational logic and registers
  - Every intermediate state is captured in a register



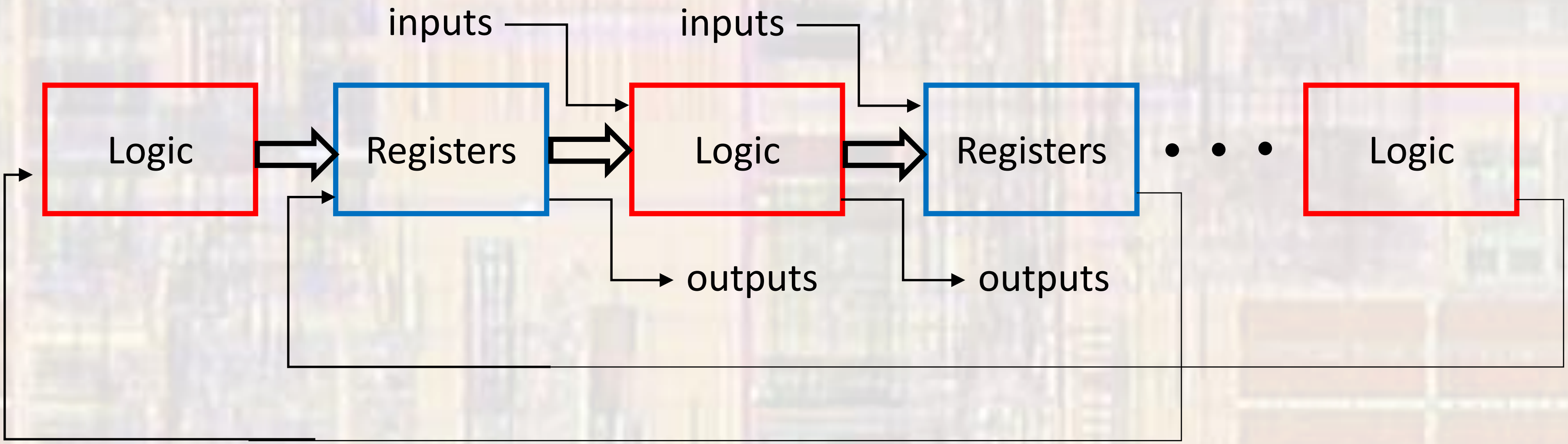
Register Transfer Architecture

- Register Transfer Architecture
  - Synchronous System
  - Blocks of combinational logic and registers
  - Every intermediate state is captured in a register
  - Many I/O paths



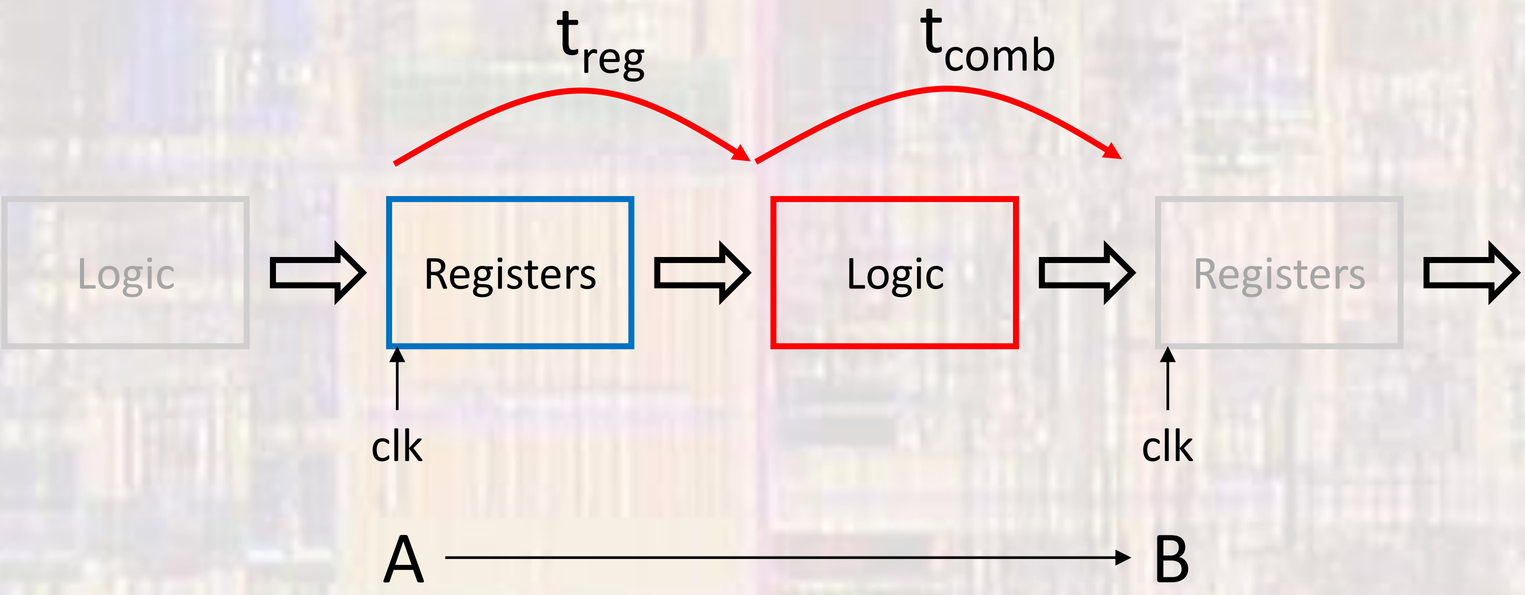
Register Transfer Architecture

- Register Transfer Architecture
  - Synchronous System
  - Blocks of combinational logic and registers
  - Every intermediate state is captured in a register
  - Many I/O paths
  - Many feedback paths



Register Transfer Architecture

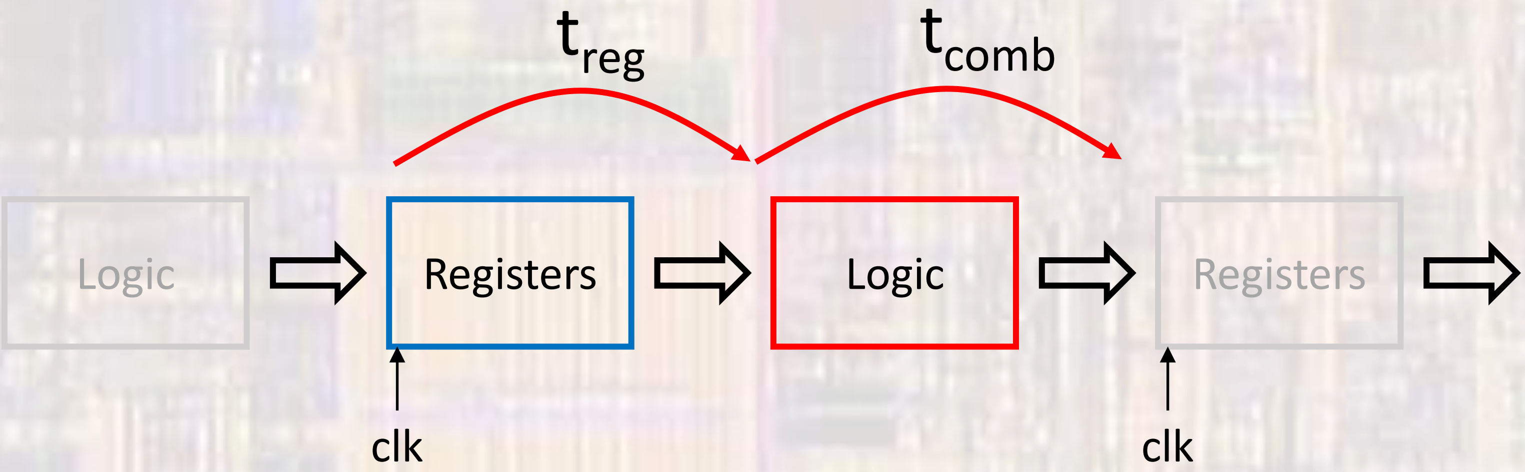
- Timing



Data must make it from A to B before the next active clock edge (rising)

Register Transfer Architecture

- Timing



$$T_{nom} = t_{CQ} + t_{comb}$$

$$T_{nom} = t_{CQ} + t_{interconnect1} + t_{comb} + t_{interconnect2} + t_{setup}$$