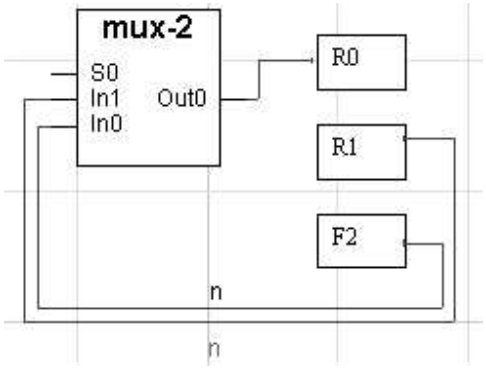
Register Transfers

June 9, 2003

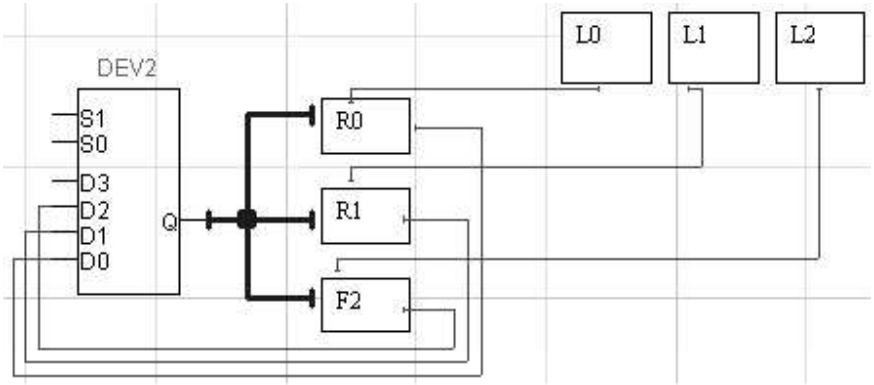
- We need to be able to take data from any register(s), do some calculations and write back to a register
- There are a few ways to do this
    - We will only worry about register transfer (RG ← R0)
    - No calculations
- Multiplexer based transfer
    - Connect a multiplexer :
        - Inputs: output from every register
        - Output: to a register input



- Works if there are a few registers—bad if there are a lot
    - 32 register: 32 31 to 1 multiplexers and a lot of connections

Bus Based transfer

- too many multiplexers would mean a large (thus costly) circuit
- we could use a single input line for all registers
    - a “bus”
    - (any shared single is a “bus”)
- the multiplexer will be connected to every input:

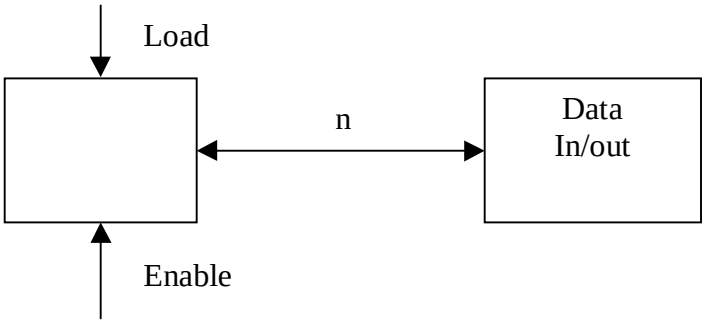


Eg. To do R0 ←R2
- select input 2 (S_1 = 1, S_0 =0)
- set L0 ← 1;

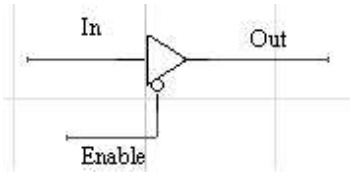
Eg. R0 ← R2, R1 ← R2
- select input 2 (S_1 = 1, S_0 =0)
- set L0 ← 1, L1 ←1
- R1 ← R0, R0 ← R1
    - Can't o
    - Could have done with the multiplexer solution

The State Bus

- We can use three-state buffer to get rid of the multiplexers all together
    - We will use a single bus for both input & output
    - A "bidirectional bus"
- We need a register with bidirectional input/output



Tri State Buffer



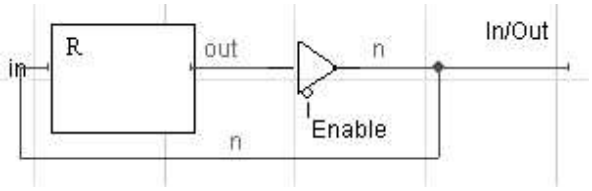
- When En is 1, it behaves like a buffer (output the input signal)
- When En is 0, don't allow any current in <u>either direction</u>

In	En	Out
0	1	0
1	1	1
0	0	Hi –Z
1	0	Hi –Z

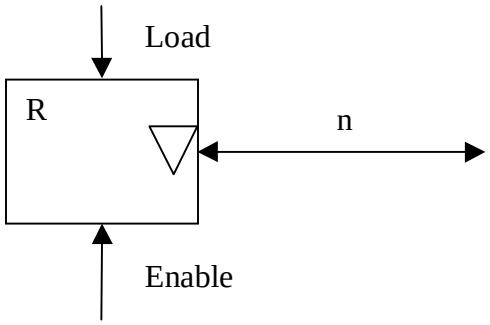
Hi-Z = high impedance
- No current allow past
- In VHDL: 'Z'

- Building a bidirectional register:

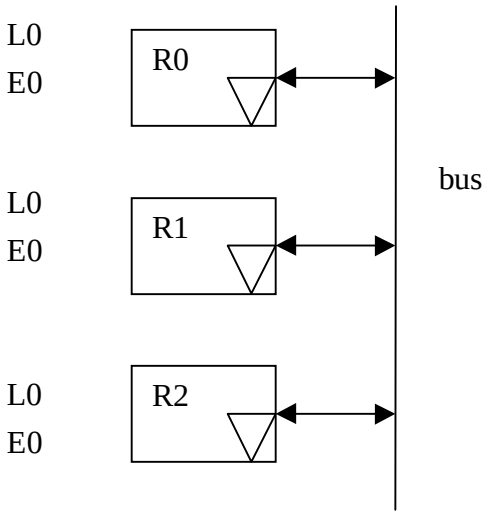
- Using a standard register:



Standard Symbol



We can now connect some bi-di-register on a bus:



At most of the enable signal can be 1 at a time

Eg. To do R0 ← R2
– set E2=1, L0 ←1
Eg. To do R0←R2 R1←R2
– E2 = 1, L0 = 1, L1= 1
Using a signal bus for input & output halves the # of connections
– Less circuitry
–