ECE 368: CAD-Based Logic Design

Lecture Notes #8

Designing and Controlling Data Paths
Basic Components, Connections and Control Operations

Shantanu Dutt
ECE Dept.
UIC

Copyright: Shantanu Dutt
(a) $2^n:1$ Mux: \( \text{sel} = j \implies \text{o/p} = L_j$

(b) $n:2^n$ DCD: \[ \text{addr} = j \land \text{enb} = 1 \implies \text{sel} = j \land \text{sel} = 0, i \neq j, \text{enb} = 0 \implies \text{sel} = 0, \text{for all } j \]

(c) Tri-state buffer:
\[ \text{enb} = 1 \implies \text{o/p} = I; \quad \text{enb} = 0 \implies \text{o/p} = Z \]
(high impedance, not connected to any signal)

(d) Bus connection via tri-state buffers: exactly one \( \text{enb} = 1 \implies \text{Bus} = L_j; \quad \text{enb} = 0, \text{for all } i, \implies \text{Bus} = Z \)
(a) +ve edge-trigg. D-FF w/o enable

(b) Timing diagram for +ve edge triggered D-FF (w/o enable)

(c) +ve edge-trigg. D-FF w/ enable

(d) 4-bit register with load/write and reset control
A) Basic Operations:

1) Load a register (individual/sepate register, e.g., the Program Counter [PC] in a processor, or one or more registers in a register file)

   e.g. \( R_1 \leftarrow \text{BusA} \) /* load reg. \( R_1 \) of BusA content */

   - To perform \( R_1 \leftarrow \text{BusA} \) needed
     the "load\( _{R1} \)" signals needs to be "activated" (generally, set to 1) by a control FSM

   - If the FSM sets load\( _{R1} = 1 \) in clock cycle \( i \), then \( R_1 \leftarrow \text{BusA} \) (i.e. loading of \( R_1 \)) is accomplished @ the +ve edge
     (assuming a +ve edge system/register) of clock cycle \( i+1 \).
2) Connecting data to a bus from one of multiple sources

E.g. Write-Bus ← Adder-output 

or Read-Bus ← register as part of 32-register file.

- One of multiple sources is selected to connect to a bus via either a mux (as shown above) or tri-state buffers, by activating the right control signals.

- In the above e.g. to accomplish Write-Bus ← Adder-output, "alu-set" control signal is set to '00' by a controller, fox, second is accomplished in the same cc in which control signal is activated.
Other combinations of the above 2 operations can be used for various types of connections and regular loadings.

E.g., loading R1 from one of multiple sources:

R1 \rightarrow \text{Bus A}

Bus A is one source out of 5 sources.

- R1 \times \text{Bus A} is now accomplished by setting values for 2 control signals `R1-input-set' and `load-R1' to 010 & 0, respectively.

- `R1-input-set' = 010

- `load-R1' = 0

- Bus A
B) Secondary Operations: 1) Register w/ various shift/rotate/load options

- oper = 00 means load-reg from D-bus (I/P Bus)
- 01 means LSR
- 10 means ASR
- 11 means ROR (rotate right)

Controller FSM:

- reset_reg1 = 1 [R1 ← 0]
- oper = 10 [R1 ← ASR(R1)]
- oper = 00 [R1 ← D-Bus]
The above shift register is really a combination of a connection to the bus of R1, from one of multiple sources, including a function unit (Fu), or a combinational circuit.

Take simple case of LSR only for R1.

**Note:** Entries at the input to R1 cell are added internally in the logic inside the R1 "box."
Similarly, other arithmetic operations like addition, subtraction, and multiplication can be accomplished by a combination of these simple basic operations. Some of the basic operations have been described already.