

CPE 426/526
Chapter 1 - Structured
Design Concepts

Dr. Rhonda Kay Gaede

UAH

1.1 The Abstraction Hierarchy

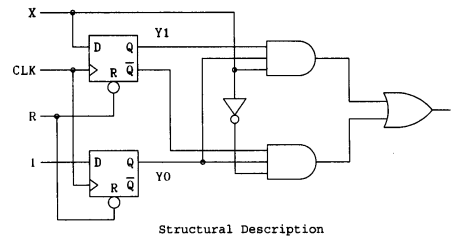
- Two Domains
 - Structural -
 - Behavioral -

1.1 The Abstraction Hierarchy - Domain Examples

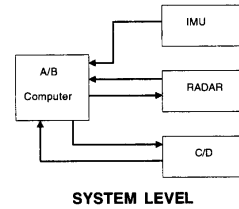
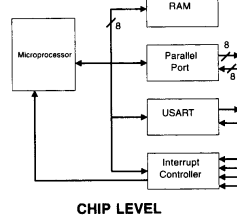
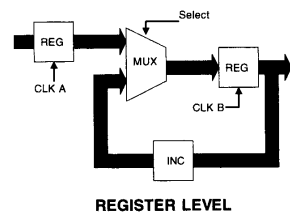
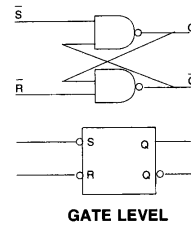
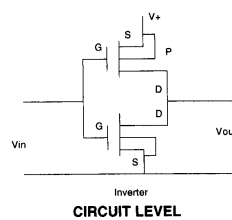
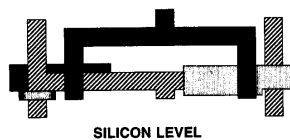
```
entity TWO_CONSECUTIVE is
  port(CLK,R,X: in BIT;Z: out BIT);
end TWO_CONSECUTIVE;
```

```
architecture DATAFLOW of TWO_CONSECUTIVE is
  signal Y1,Y0: BIT;
begin
  STATE: block((CLK = '1'and not CLK'STABLE) or R = '0')
  begin
    Y1 <= guarded '0' when R = '0' else X;
    Y0 <= guarded '0' when R = '0' else '1';
  end block STATE;
  Z <= Y0 and ((not Y1 and not X) or (Y1 and X));
end DATAFLOW;
```

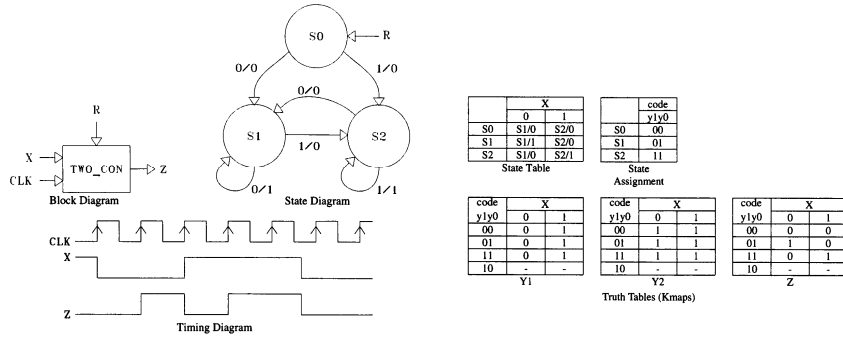
-- Behavioral description



1.1 The Abstraction Hierarchy - Book Design Hierarchy



1.2 Textual vs. Pictorial Representations



- A hardware description language is an enhanced high-level programming language.

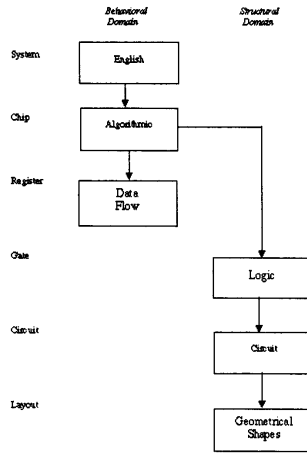
1.3 Types of Behavioral Descriptions

```
entity TWO_CONSECUTIVE is
    port(CLK,R,X: in BIT;Z: out BIT);
end TWO_CONSECUTIVE;

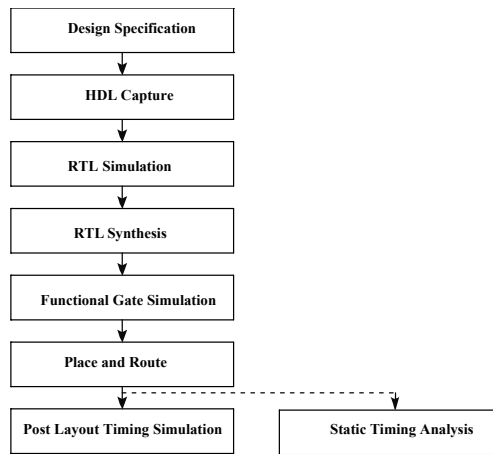
architecture DATAFLOW of TWO_CONSECUTIVE
    is
        signal Y1,Y0: BIT;
    begin
        STATE: block((CLK = '1'and not
            CLK'STABLE) or R = '0')
        begin
            Y1 <= guarded '0' when R = '0' else
                X;
            Y0 <= guarded '0' when R = '0' else
                '1';
        end block STATE;
        Z <= Y0 and ((not Y1 and not X) or (Y1
            and X));
    end DATAFLOW;
```

```
architecture ALGORITHMIC of TWO_CONSECUTIVE is
    type STATE is (S0,S1,S2);
    signal Q: STATE := S0;
begin
    process(R,X,CLK,Q)
    begin
        if (R'EVENT and R = '0') then --reset
            event
            Q <= S0;
        elsif (CLK'EVENT and CLK = '1') then
            if X = '0' then
                Q <= S1;
            else
                Q <= S2;
            end if;
        end if;
        if Q'EVENT or X'EVENT then --output
            function
            if (Q=S1 and X='0') or (Q=S2 and X='1')
            then
                Z <= '1';
            else
                Z <= '0';
            end if;
        end if;
    end process;
end ALGORITHMIC;
```

1.4 Design Process - Book View



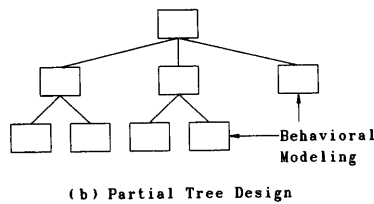
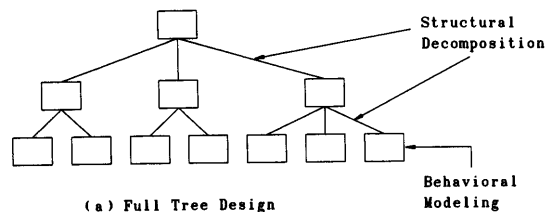
1.4 Design Process - Alternate View



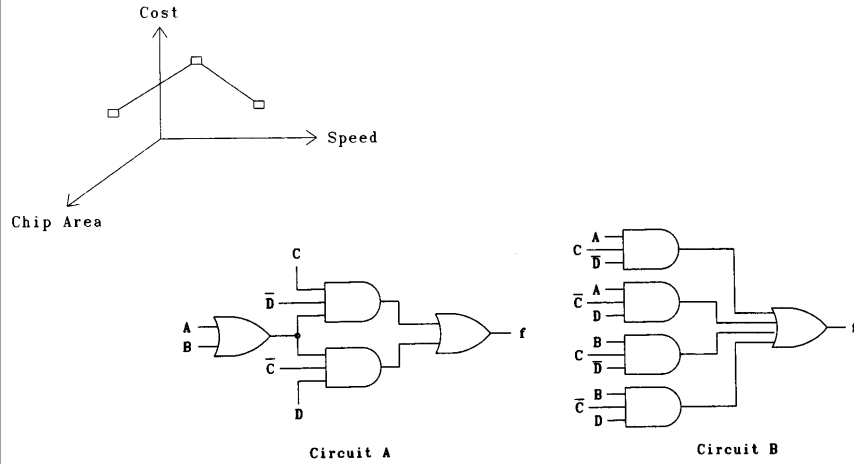
1.4 Design Process - Types of Synthesis

- English → algorithmic representation
- Algorithmic → data flow
- Algorithmic → gate level
- Data flow → gate level
- Gate level → layout

1.5 Structural Design Decomposition



1.6 The Digital Design Space



The Big Picture

