

Chapter 1 Solutions¹

1.1.

General-purpose microprocessors: 1GHz to 3GHz.
Dedicated microprocessors: 4MHz to 50MHz.

1.7.

CPU	Year Introduced	Clock Speed	Number of Transistors
8086	1978	4.7 – 10 MHz	29,000
80286	1982	6 – 12 MHz	134,000
80386	1985	16 – 33 MHz	275,000
80486	1989	25 – 100 MHz	1.2 million
Pentium	1993	60 – 200 MHz	3.3 million
Pentium Pro	1995	150 – 200 MHz	5.5 million
Pentium II	1997	234 – 450 MHz	7.5 million
Celeron	1998	266 – 800 MHz	19 million
Pentium III	1999	400 MHz – 1.2 GHz	28 million
Pentium 4	2000	1.4 – 3 GHz	42 million

1.8.

The Intel Core i7 (Quad) CPU has about 731,000,000 transistors.
The Xilinx Virtex-7 FPGA has about 6,800,000,000 transistors.
The Altera Stratix V FPGA has about 3,800,000,000 transistors.
Reference: http://en.wikipedia.org/wiki/Transistor_count

1.9.

```
module multiplexer (  
    input s, d0, d1,  
    output y  
);  
  
    assign y = (~s & ~d1 & d0) | (~s & d1 & d0) | (s & d1 & ~d0) |  
              (s & d1 & d0);  
  
endmodule
```

1.10.

```
module multiplexer (  
    input s, d0, d1,  
    output y  
);  
  
    wire sn,d1n,d0n,a1,a2,a3,a4;  
  
    // first parameter is the output; remaining parameters are the inputs  
    not U0(sn,s);
```

¹ Note that many of the solutions are not unique and that there are other correct answers.

```

    not U1(d1n,d1);
    not U2(d0n,d0);
    and U3(a1,sn,d1n,d0);
    and U4(a2,sn,d1,d0);
    and U5(a3,s,d1,d0n);
    and U6(a4,s,d1,d0);
    or  U7(y,a1,a2,a3,a4);

endmodule

```

1.11.

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY multiplexer IS PORT(
    d0, d1, s: IN STD_LOGIC;
    y: OUT STD_LOGIC);
END multiplexer;

ARCHITECTURE Dataflow OF multiplexer IS
BEGIN
    y <= ((NOT s) AND (NOT d1) AND d0) OR ((NOT s) AND d1 AND d0) OR
        (s AND d1 AND (NOT d0)) OR (s AND d1 AND d0);
END Dataflow;

```

1.12.

```

----- NOT gate -----
LIBRARY IEEE
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY notgate IS PORT(
    i: IN STD_LOGIC;
    o: OUT STD_LOGIC);
END notgate;
ARCHITECTURE Dataflow OF notgate IS
BEGIN
    o <= not i;
END Dataflow;

----- 3-input AND gate -----
LIBRARY IEEE
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY and3gate IS PORT(
    i1, i2, i3: IN STD_LOGIC;
    o: OUT STD_LOGIC);
END and3gate;
ARCHITECTURE Dataflow OF and3gate IS
BEGIN
    o <= i1 AND i2 AND i3;
END Dataflow;

----- 4-input OR gate -----
LIBRARY IEEE
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY or4gate IS PORT(

```

```

    i1, i2, i3, i4: IN STD_LOGIC;
    o: OUT STD_LOGIC);
END or4gate;
ARCHITECTURE Dataflow OF or4gate IS
BEGIN
    o <= i1 OR i2 OR i3 OR i4;
END Dataflow;

----- 2-to-1 multiplexer -----
LIBRARY IEEE
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY multiplexer IS PORT(
    d0, d1, s: IN STD_LOGIC;
    y: OUT STD_LOGIC);
END multiplexer;
ARCHITECTURE Structural OF multiplexer IS
    COMPONENT notgate PORT(
        i: IN STD_LOGIC;
        o: OUT STD_LOGIC);
    END COMPONENT;
    COMPONENT and3gate PORT(
        i1, i2, i3: IN STD_LOGIC;
        o: OUT STD_LOGIC);
    END COMPONENT;
    COMPONENT or4gate PORT(
        i1, i2, i3, i4: IN STD_LOGIC;
        o: OUT STD_LOGIC);
    END COMPONENT;

    SIGNAL sn, d0n, d1n, and1, and2, and3, and4: STD_LOGIC;

BEGIN
    U1: notgate PORT MAP(s, sn);
    U2: notgate PORT MAP(d0, d0n);
    U3: notgate PORT MAP(d1, d1n);
    U4: and3gate PORT MAP(sn, d1n, d0, and1);
    U5: and3gate PORT MAP(sn, d1, d0, and2);
    U6: and3gate PORT MAP(s, d1, d0n, and3);
    U7: and3gate PORT MAP(s, d1, d0, and4);
    U8: or4gate PORT MAP(and1, and2, and3, and4, y);
END Structural;

```

