Computer Architecture - Project 1A
Datapath Design

Vsevolod Syrtsov
syrtsovv@tcd.ie
source: github.com/futurecertificate
Lecturer: Dr. Michael Manzke

Contents

1 VHDL Sources

1.1 16 bit Register

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity reg16 is
port (
D : in std_logic_vector(15 downto 0);
load, Clk : in std_logic;
Q : out std_logic_vector(15 downto 0)
);

end reg16;
architecture Behavioral of reg16 is
begin
process(Clk)
begin
if (rising_edge(Clk)) then
if load='1' then
Q<=D after 5 ns;
end if;
end if;
end process;
end Behavioral;
```

1.2 Decoder 3 to 8

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity decoder3_8 is
Port ( A0 : in std_logic;
A1 : in std_logic;
A2 : in std_logic;

Q0 : out std_logic;
Q1 : out std_logic;
```

```
Q2 : out std_logic;
Q3 : out std_logic;
Q4 : out std_logic;
Q5 : out std_logic;
Q6 : out std_logic;
Q7 : out std_logic);

end decoder3_8;
architecture Behavioral of decoder3_8 is
begin
Q0<= ((not A0) and (not A1) and (not A2)) after 5 ns;
Q1<= ((not A0) and (not A1) and A2) after 5 ns;
Q2<= ((not A0) and A1 and (not A2)) after 5 ns;
Q3<= ((not A0) and A1 and A2) after 5 ns;
Q4<= (A0 and (not A1) and (not A2)) after 5 ns;
Q5<= (A0 and (not A1) and A2) after 5 ns;
Q6<= (A0 and A1 and (not A2)) after 5 ns;
Q7<= (A0 and A1 and A2) after 5 ns;

end Behavioral;
```

1.3 Multiplexer 2 to 16

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity mux2_16 is
port (
In0 : in std_logic_vector(15 downto 0);
In1 : in std_logic_vector(15 downto 0);
s : in std_logic;
Z : out std_logic_vector(15 downto 0));

end mux2_16;
architecture Behavioral of mux2_16 is
begin
Z <= In0 after 5 ns when S='0' else
In1 after 5 ns when S='1'else
"0000000000000000" after 5 ns;
```

```
end Behavioral;
```

1.4 Multiplexer 8 to 16

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity mux8_16 is
Port ( In0, In1, In2, In3, In4, In5, In6, In7 : in std_logic_vector(15 downto 0)
S0, S1, S2 : in std_logic;
Z : out std_logic_vector(15 downto 0));

end mux8_16;
architecture Behavioral of mux8_16 is
begin
Z <= In0 after 5 ns when S0='0' and S1='0' and S2 ='0' else
In1 after 5 ns when S0='0' and S1='0' and S2 ='1' else
In2 after 5 ns when S0='0' and S1='1' and S2 ='0' else
In3 after 5 ns when S0='0' and S1='1' and S2 ='1' else
In4 after 5 ns when S0='1' and S1='0' and S2 ='0' else
In5 after 5 ns when S0='1' and S1='0' and S2 ='1' else
In6 after 5 ns when S0='1' and S1='1' and S2 ='1' else
In7 after 5 ns when S0='1' and S1='1' and S2 ='1' else
"0000000000000000" after 5 ns;
end Behavioral;
```

1.5 Register File

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity reg_file is
Port (
src_s0 : in std_logic;
src_s1 : in std_logic;
src_s2 : in std_logic;

des_A0 : in std_logic;
des_A1 : in std_logic;
```

```
des_A2 : in std_logic;

Clk : in std_logic;
data_src : in std_logic;

data : in std_logic_vector(15 downto 0);
reg0 : out std_logic_vector(15 downto 0);
reg1 : out std_logic_vector(15 downto 0);
reg2 : out std_logic_vector(15 downto 0);
reg3 : out std_logic_vector(15 downto 0);
reg4 : out std_logic_vector(15 downto 0);
reg5 : out std_logic_vector(15 downto 0);
reg6 : out std_logic_vector(15 downto 0);
reg7 : out std_logic_vector(15 downto 0)
);

end reg_file;

architecture Behavioral of reg_file is

COMPONENT reg16
PORT(

D : IN std_logic_vector(15 downto 0);
load : IN std_logic;
Clk : IN std_logic;
Q : OUT std_logic_vector(15 downto 0)
);

END COMPONENT;

COMPONENT mux8_16
PORT(

In0 : IN std_logic_vector(15 downto 0);
In1 : IN std_logic_vector(15 downto 0);
In2 : IN std_logic_vector(15 downto 0);
In3 : IN std_logic_vector(15 downto 0);
In4 : IN std_logic_vector(15 downto 0);
In5 : IN std_logic_vector(15 downto 0);
In6 : IN std_logic_vector(15 downto 0);
```

```
In7 : IN std_logic_vector(15 downto 0);

S0 : IN std_logic;
S1 : IN std_logic;
S2 : IN std_logic;
Z : OUT std_logic_vector(15 downto 0)
);
END COMPONENT;

COMPONENT mux2_16
PORT(

In0 : IN std_logic_vector(15 downto 0);
In1 : IN std_logic_vector(15 downto 0);
s : IN std_logic;
Z : OUT std_logic_vector(15 downto 0)
);
END COMPONENT;

COMPONENT decoder3_8
PORT(

A0 : IN std_logic;
A1 : IN std_logic;
A2 : IN std_logic;
Q0 : OUT std_logic;
Q1 : OUT std_logic;
Q2 : OUT std_logic;
Q3 : OUT std_logic;
Q4 : OUT std_logic;
Q5 : OUT std_logic;
Q6 : OUT std_logic;
Q7 : OUT std_logic
);
END COMPONENT;

signal load_reg0, load_reg1, load_reg2, load_reg3, load_reg4, load_reg5, load_re
signal reg0_q, reg1_q, reg2_q, reg3_q, reg4_q, reg5_q, reg6_q, reg7_q, data_src_


begin
```

```
reg00: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg0,
Clk => Clk,
Q => reg0_q

);

reg01: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg1,
Clk => Clk,
Q => reg1_q

);

reg02: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg2,
Clk => Clk,
Q => reg2_q

);

reg03: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg3,
Clk => Clk,
Q => reg3_q

);

reg04: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg4,
```

```
Clk => Clk,
Q => reg4_q

);

reg05: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg5,
Clk => Clk,
Q => reg5_q

);

reg06: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg6,
Clk => Clk,
Q => reg6_q

);

reg07: reg16 PORT MAP(

D => data_src_mux_out,
load => load_reg7,
Clk => Clk,
Q => reg7_q

);

decoder_3_8: decoder3_8 PORT MAP(

A0 => des_A0,
A1 => des_A1,
A2 => des_A2,

Q0 => load_reg0,
Q1 => load_reg1,
Q2 => load_reg2,
```

```
Q3 => load_reg3,
Q4 => load_reg4,
Q5 => load_reg5,
Q6 => load_reg6,
Q7 => load_reg7

);

data_src_mux2_16bit: mux2_16 PORT MAP(

In0 => data,
In1 => src_reg,
s => data_src,
Z => data_src_mux_out

);

Inst_mux2_16bit: mux8_16 PORT MAP(

In0 => reg0_q,
In1 => reg1_q,
In2 => reg2_q,
In3 => reg3_q,
In4 => reg4_q,
In5 => reg5_q,
In6 => reg6_q,
In7 => reg7_q,

S0 => src_s0,
S1 => src_s1,
S2 => src_s2,
Z => src_reg

);
reg0 <= reg0_q;
reg1 <= reg1_q;
reg2 <= reg2_q;
reg3 <= reg3_q;
reg4 <= reg4_q;
reg5 <= reg5_q;
reg6 <= reg6_q;
```

```
reg7 <= reg7_q;

end Behavioral;
```

2 Component Test Benches

2.1 Register Testbench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;


ENTITY tbreg16 IS
END tbreg16;

ARCHITECTURE behavior OF tbreg16 IS


    COMPONENT reg16
    PORT(
         D : IN  std_logic_vector(15 downto 0);
         load : IN  std_logic;
         clk : IN  std_logic;
         Q : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   signal D : std_logic_vector(15 downto 0) := (others => '0');
   signal load : std_logic := '0';
   signal clk : std_logic := '0';

         signal Q : std_logic_vector(15 downto 0);


BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: reg16 PORT MAP (
          D => D,
```

```
        load => load,
        clk => clk,
        Q => Q
      );

   -- Clock process definitions
   clk_process :process
   begin
                clk <= '0';
                wait for 5ns;
                clk <= '1';
                wait for 5ns;
   end process;

   -- Stimulus process
   stim_proc: process
   begin
                wait for 10ns;
                D <= x"1234";
                load <= '1';

                wait for 10ns;
                load <= '0';

                wait for 10ns;
                D <= x"FFFF";
                load <= '1';

                wait for 10ns;
                load <= '0';
   end process;

END;
```

2.2 Decoder 3 to 8

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity tb3_8dec is
```

```
end tb3_8dec;

architecture Behavioral of tb3_8dec is
COMPONENT decoder3_8
    PORT(
         A0 : IN  std_logic;
         A1 : IN  std_logic;
         A2 : IN  std_logic;

         Q0 : out  std_logic;
         Q1 : out  std_logic;
         Q2 : out  std_logic;
         Q3 : out  std_logic;
         Q4 : out  std_logic;
         Q5 : out  std_logic;
         Q6 : out  std_logic;
         Q7 : out  std_logic
        );
    END COMPONENT;

     --Inputs

   signal A0 : std_logic := '0';
   signal A1 : std_logic := '0';
   signal A2 : std_logic := '0';

         --Outputs
   signal Q0 : std_logic;
   signal Q1 : std_logic;
   signal Q2 : std_logic;
   signal Q3 : std_logic;
   signal Q4 : std_logic;
   signal Q5 : std_logic;
   signal Q6 : std_logic;
   signal Q7 : std_logic;

begin
--Unit Under Testing
uut: decoder3_8 PORT MAP (
          Q0 => Q0,
```

```
            Q1 => Q1,
            Q2 => Q2,
            Q3 => Q3,
            Q4 => Q4,
            Q5 => Q5,
            Q6 => Q6,
            Q7 => Q7,

            A0 => A0,
            A1 => A1,
            A2 => A2

        );
stim_proc: process
    begin
                wait for 10ns;
                A0 <= '0';
                A1 <= '0';
                A2 <= '0';


                wait for 10ns;
                A0 <= '0';
                A1 <= '0';
                A2 <= '1';


                wait for 10ns;
                A0 <= '0';
                A1 <= '1';
                A2 <= '0';


                wait for 10ns;
                A0 <= '0';
                A1 <= '1';
                A2 <= '1';

            wait for 10ns;
                A0 <= '1';
                A1 <= '0';
```

```
                A2 <= '0';


                wait for 10ns;
                A0 <= '1';
                A1 <= '0';
                A2 <= '1';


                wait for 10ns;
                A0 <= '1';
                A1 <= '1';
                A2 <= '0';


                wait for 10ns;
                A0 <= '1';
                A1 <= '1';
                A2 <= '1';


    end process;
end Behavioral;
```

2.3 Multiplexer 2 to 16

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;


ENTITY tb2_16 IS
END tb2_16;

ARCHITECTURE behavior OF tb2_16 IS


    COMPONENT mux2_16
    PORT(
         s : IN  std_logic;
         in0 : IN  std_logic_vector(15 downto 0);
         in1 : IN  std_logic_vector(15 downto 0);
```

```
         z : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   signal in0 : std_logic_vector(15 downto 0) := (others => '0');
   signal in1 : std_logic_vector(15 downto 0) := (others => '0');
   signal s : std_logic := '0';

   signal z : std_logic_vector(15 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

--   constant Clk_period : time := 10 ns;

BEGIN

        -- Instantiate the Unit Under Test (UUT)
    uut: mux2_16 PORT MAP (
          in0 => in0,
          in1 => in1,
          s => s,
          z => z
        );

    stim_proc: process
    begin
                wait for 10ns;
                in0 <= x"BEEF";
                in1 <= x"B00B";

                wait for 10ns;
                s <= '1';

                wait for 10ns;
                s <= '0';

                wait for 10ns;
                s <= '1';

    end process;
```

```
END;
```

2.4 Multiplexer 8 to 16

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY tb8_16 IS
END tb8_16;

ARCHITECTURE behavior OF tb8_16 IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mux8_16
    PORT(
         s0 : IN  std_logic;
         s1 : IN  std_logic;
         s2 : IN  std_logic;

         in0 : IN  std_logic_vector(15 downto 0);
         in1 : IN  std_logic_vector(15 downto 0);
         in2 : IN  std_logic_vector(15 downto 0);
         in3 : IN  std_logic_vector(15 downto 0);
         in4 : IN  std_logic_vector(15 downto 0);
         in5 : IN  std_logic_vector(15 downto 0);
         in6 : IN  std_logic_vector(15 downto 0);
         in7 : IN  std_logic_vector(15 downto 0);
         z : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal in0 : std_logic_vector(15 downto 0) := (others => '0');
   signal in1 : std_logic_vector(15 downto 0) := (others => '0');
```

```
   signal in2 : std_logic_vector(15 downto 0) := (others => '0');
   signal in3 : std_logic_vector(15 downto 0) := (others => '0');
   signal in4 : std_logic_vector(15 downto 0) := (others => '0');
   signal in5 : std_logic_vector(15 downto 0) := (others => '0');
   signal in6 : std_logic_vector(15 downto 0) := (others => '0');
   signal in7 : std_logic_vector(15 downto 0) := (others => '0');

   signal s0 : std_logic := '0';
   signal s1 : std_logic := '0';
   signal s2 : std_logic := '0';

         --Outputs
   signal z : std_logic_vector(15 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

--   constant Clk_period : time := 10 ns;

BEGIN

        -- Instantiate the Unit Under Test (UUT)
    uut: mux8_16 PORT MAP (
          in0 => in0,
          in1 => in1,
          in2 => in2,
          in3 => in3,
          in4 => in4,
          in5 => in5,
          in6 => in6,
          in7 => in7,

          s0 => s0,
          s1 => s1,
          s2 => s2,

          z => z
        );

    stim_proc: process
    begin
                wait for 10ns;
```

```
                in0 <= x"BEEF";
                in1 <= x"B00B";
                in2 <= x"BAAA";
                in3 <= x"BABE";
                in4 <= x"FAB0";
                in5 <= x"BADD";
                in6 <= x"DAD3";
                in7 <= x"CA75";


                wait for 10ns;
                s0 <= '0';
                s1 <= '0';
                s2 <= '0';


                wait for 10ns;
                s0 <= '1';
                s1 <= '1';
                s2 <= '1';


                wait for 10ns;
                s0 <= '0';
                s1 <= '0';
                s2 <= '0';


    end process;

END;
```

2.5 Register File

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;


ENTITY tbreg_file IS
END tbreg_file;
```

```
ARCHITECTURE behavior OF tbreg_file IS

    COMPONENT reg_file
    PORT(
        src_s0 : in std_logic;
        src_s1 : in std_logic;
        src_s2 : in std_logic;

        des_A0 : in std_logic;
        des_A1 : in std_logic;
        des_A2 : in std_logic;

        Clk : in std_logic;
        data_src : in std_logic;

        data : in std_logic_vector(15 downto 0);
        reg0 : out std_logic_vector(15 downto 0);
        reg1 : out std_logic_vector(15 downto 0);
        reg2 : out std_logic_vector(15 downto 0);
        reg3 : out std_logic_vector(15 downto 0);
        reg4 : out std_logic_vector(15 downto 0);
        reg5 : out std_logic_vector(15 downto 0);
        reg6 : out std_logic_vector(15 downto 0);
        reg7 : out std_logic_vector(15 downto 0)
        );
    END COMPONENT;


        --Inputs
        signal src_s0 : std_logic := '0';
        signal src_s1 : std_logic := '0';
        signal src_s2 : std_logic := '0';

        signal des_A0 : std_logic := '0';
        signal des_A1 : std_logic := '0';
        signal des_A2 : std_logic := '0';

        signal data_src : std_logic := '0';
        signal clk : std_logic := '0';
        signal data : std_logic_vector(15 downto 0) := (others => '0');
```

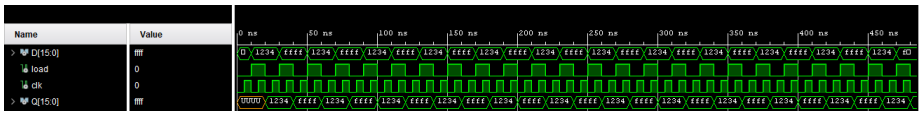
```
        --Outputs
        signal reg0 : std_logic_vector(15 downto 0);
        signal reg1 : std_logic_vector(15 downto 0);
        signal reg2 : std_logic_vector(15 downto 0);
        signal reg3 : std_logic_vector(15 downto 0);
        signal reg4 : std_logic_vector(15 downto 0);
        signal reg5 : std_logic_vector(15 downto 0);
        signal reg6 : std_logic_vector(15 downto 0);
        signal reg7 : std_logic_vector(15 downto 0);

BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: reg_file PORT MAP (
          src_s0 => src_s0,
          src_s1 => src_s1,
          src_s2 => src_s2,

          des_A0 => des_A0,
          des_A1 => des_A1,
          des_A2 => des_A2,

          data_src => data_src,
          Clk => clk,
          data => data,

          reg0 => reg0,
          reg1 => reg1,
          reg2 => reg2,
          reg3 => reg3,
          reg4 => reg4,
          reg5 => reg5,
          reg6 => reg6,
          reg7 => reg7

        );

   -- Clock process definitions
   clk_process :process
   begin
```

```
                clk <= '0';
                wait for 5ns;
                clk <= '1';
                wait for 5ns;
end process;

-- Stimulus process
stim_proc: process
begin
                wait for 10ns;
                des_a0 <= '0';
                des_a1 <= '0';
                des_a2 <= '0';
                data <= x"0000";

                wait for 10ns;
                des_a0 <= '0';
                des_a1 <= '0';
                des_a2 <= '1';
                data <= x"1111";

                wait for 10ns;
                des_a0 <= '0';
                des_a1 <= '1';
                des_a2 <= '0';
                data <= x"2222";

                wait for 10ns;
                des_a0 <= '0';
                des_a1 <= '1';
                des_a2 <= '1';
                data <= x"3333";

                wait for 10ns;
                des_a0 <= '1';
                des_a1 <= '0';
                des_a2 <= '0';
                data <= x"4444";

                wait for 10ns;
                des_a0 <= '1';
```

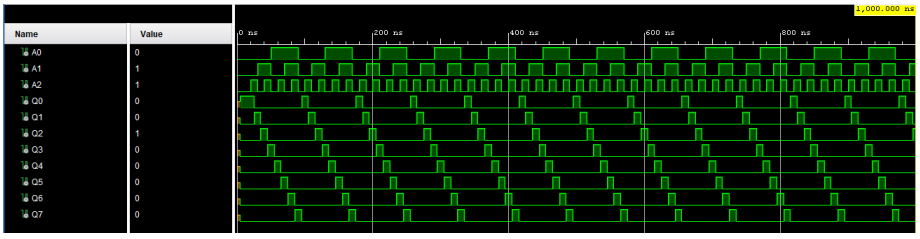
```
                des_a1 <= '0';
                des_a2 <= '1';
                data <= x"5555";

                wait for 10ns;
                des_a0 <= '1';
                des_a1 <= '1';
                des_a2 <= '0';
                data <= x"6666";

                wait for 10ns;
                des_a0 <= '1';
                des_a1 <= '1';
                des_a2 <= '1';
                data <= x"7777";

    end process;

END;
```

3 Testbench Results

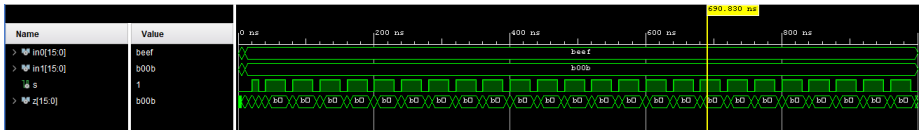
3.1 Register



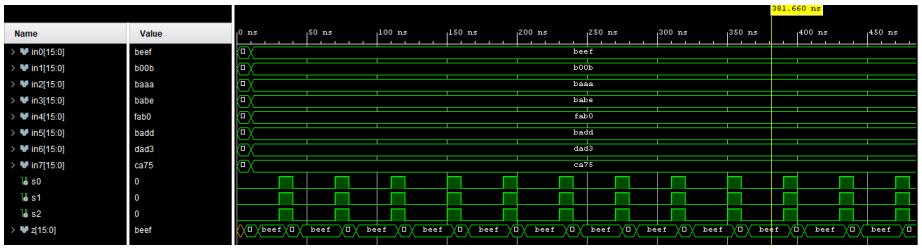
3.2 Decoder



3.3 2-16 Multiplexer



3.4 8-16 Multiplexer



3.5 Register File