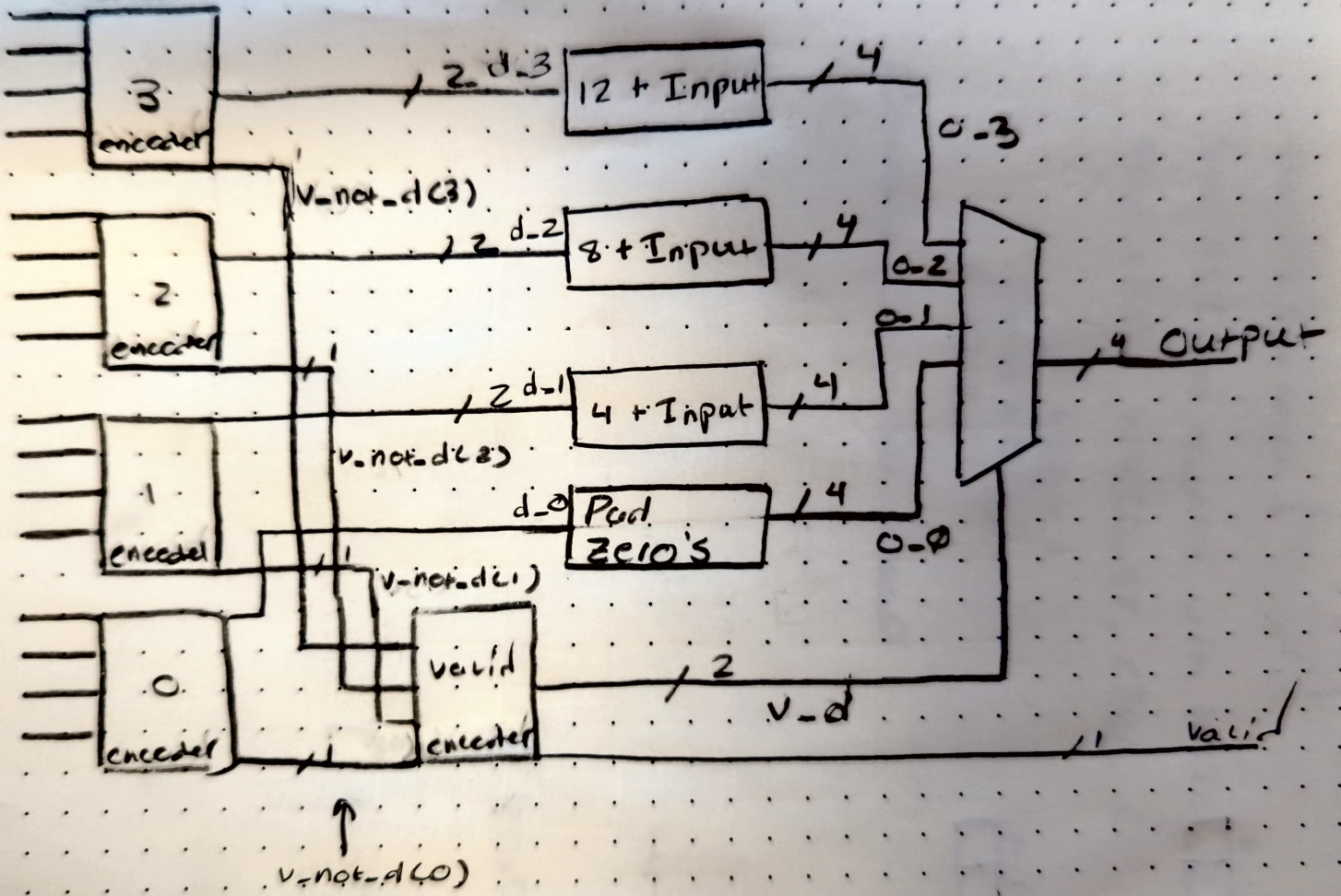
3 encoder
2 encoder
1 encoder
0 encoder
2 d_3
12 + Input
4
O_3
v_not_d(3)
2 d_2
8 + Input
O_2
O_1
4 output
1
2 d_1
4 + Inpat
v_not_d(2)
d_0
Pad Zero's
O_0
v_not_d(1)
valid encoder
2
V_d
valid
v_not_d(0)

```vhdl
-- VHDL 8 bit processor project
-- 16 x 4 encoder
-- James Hicks Oct 15 2023

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity encoder_16x4 is
  port(r : in std_logic_vector (15 downto 0);
       e : out std_logic_vector (3 downto 0);
       v : out std_logic);
end encoder_16x4;

architecture my_arch of encoder_16x4 is

  signal d_3, d_2, d_1, d_0, v_d : std_logic_vector(1 downto 0);
  signal o_3, o_2, o_1, o_0, v_not_d : std_logic_vector(3 downto 0);
  signal twelve, eight, four : unsigned(3 downto 0);

begin

  dec_3 : entity work.encoder_4x2(cond_arch)
   port map (
     r => r (15 downto 12),
     e => d_3,
     v => v_not_d(3)
     );

  dec_2 : entity work.encoder_4x2(sel_arch)
   port map (
     r => r (11 downto 8),
     e => d_2,
     v => v_not_d(2)
     );

  dec_1 : entity work.encoder_4x2(if_arch)
   port map (
     r => r (7 downto 4),
     e => d_1,
     v => v_not_d(1)
     );

  dec_0 : entity work.encoder_4x2(case_arch)
   port map (
     r => r (3 downto 0),
     e => d_0,
     v => v_not_d(0)
     );

  valid_decoder : entity work.encoder_4x2(sel_arch)
    port map(
      r => v_not_d,
      e => v_d,
      v => v
      );

  twelve <= "1100";
  eight <= "1000";
  four <= "0100";

  with v_d select e <=
    std_logic_vector(twelve + unsigned(d_3)) when "11",
    std_logic_vector(eight + unsigned(d_2)) when "10",
    std_logic_vector(four + unsigned(d_1)) when "01",
    std_logic_vector("00" & d_0) when "00",
    "0000" when others;

end my_arch;

```

```vhdl
-- VHDL 8 bit processor project
-- 4 x 2 encoder
-- James Hicks Oct 15 2023

library ieee;
use ieee.std_logic_1164.all;

entity encoder_4x2 is
  port(r : in std_logic_vector(3 downto 0);
       e : out std_logic_vector(1 downto 0);
       v : out std_logic);
end encoder_4x2;

architecture cond_arch of encoder_4x2 is
begin

  v <= '0' when r = "0000" else
       '1';

  e <= "11" when r(3) = '1' else
       "10" when r(2) = '1' else
       "01" when r(1) = '1' else
       "00";

end cond_arch;

architecture sel_arch of encoder_4x2 is
begin

  v <= '1' when r /= "0000" else
       '0';

  with r select (e) <=
    "11" when "1000" | "1001" | "1010" | "1011" |
              "1100" | "1101" | "1110" | "1111",
    "10" when "0100" | "0101" | "0110" | "0111",
    "01" when "0010" | "0011",
    "00" when others;

end sel_arch;

architecture if_arch of encoder_4x2 is
begin

  process(r)
  begin
    if r(3) = '1' then
      e <= "11";
      v <= '1';
    elsif r(2) = '1' then
      e <= "10";
      v <= '1';
    elsif r(1) = '1' then
      e <= "01";
      v <= '1';
    elsif r(0) = '1' then
      e <= "00";
      v <= '1';
    else
      e <= "00";
      v <= '0';
    end if;
  end process;

end if_arch;

architecture case_arch of encoder_4x2 is
begin

  process(r)

  begin
```

```
    case r is
      when "1000" | "1001" | "1010" | "1011" |
           "1100" | "1101" | "1110" | "1111" =>
        e <= "11";
        v <= '1';
      when  "0100" | "0101" | "0110" | "0111" =>
        e <= "10";
        v <= '1';
      when "0010" | "0011" =>
        e <= "01";
        v <= '1';
      when "0001" =>
        e <= "00";
        v <= '1';
      when others =>
        e <= "00";
        v <= '0';
      end case;
  end process;

end case_arch;

```

James Hlavs

Lab 4: Your Name

Instructor signature

Deliverables: