1. A description of the data being handled, including data type and bit-size.
   1. The combinational components here are meant to handle inputs of at least 16 bits. For example, the adder takes in 2 16-bit inputs and returns a 16-bit result. Some components serve to only help other components. This can be seen in how the half adder and full adder are not used on their own but are used in the addition and subtraction modules.
2. List of Components
   1. Half Adder
      1. Name of Verilog Module: Add\_half
      2. Inputs
         1. a -- has bit size 1
         2. b -- had bit size 1
      3. Outputs
         1. c\_out -- has bit size 1
         2. sum -- has bit size 1
      4. Interfaces
         1. No interfaces are used in this component.
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. This component is a half-adder which takes in 2 1-bit inputs and returns the value of their sum and c\_out, which are both 1-bit as well.
      7. Documentation: Retrieved from the provided code on HW 2, provided by Dr. Becker
   2. Full Adder
      1. Name of Verilog Module: Add\_full
      2. Inputs
         1. a -- has bit size 1
         2. b -- had bit size 1
         3. c\_in -- has bit size 1
      3. Outputs
         1. c\_out -- has bit size 1
         2. sum -- has bit size 1
      4. Interfaces
         1. The full adder uses two half adders meaning we need the following wires as interfaces:
            1. Wire w1 serves as the c\_out of the first half adder and is fed into an or gate as an input to calculate the c\_out of the full adder.
            2. Wire w2 serves as the sum of the first half adder and is fed into the second half adder as input a.
            3. Wire w3 serves as the c\_out of the second half adder and is fed into an or gate as an input to calculate the c\_out of the full adder
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. This component is a full-adder which takes in 2 1-bit inputs as well as a carry-in bit and returns the value of their sum and c\_out, which are both 1-bit as well.
      7. Documentation: Retrieved from the provided code on HW 2, provided by Dr. Becker
   3. Addition, 16-bit
      1. Name of Verilog Module: adder16
      2. Inputs
         1. a -- has 16 bits
         2. b -- has 16 bits
      3. Outputs
         1. c -- has 17 bits
         2. s -- has 16 bits
         3. ovf -- has bit size 1
      4. Interfaces
         1. The 16 bit adder uses 16 full adders and we used the following wires as interfaces:
            1. Wire d serves to store the result of ANDing two 0s and gets input as the carry in of the first full adder.
            2. Wire v serves to store the result of applying XNOR on the most significant bits of inputs a and b. It is then fed into an AND gate as an input to determine if overflow occurred.
            3. Wire w serves to store the result of applying XOR on the most significant bits of input b and output s. It is then fed into an AND gate as an input to determine if overflow occurred.
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. This component takes in 2 16-bit inputs and adds them together and displays the sum as a 16-bit output. It also displays if overflow occurred because of the addition.
      7. Documentation: Written by Prathyusha Thiruvuri
   4. Subtraction, 16-bit
      1. Name of Verilog Module: subtractor16
      2. Inputs
         1. a -- has 16 bits
         2. b -- has 16 bits
      3. Outputs
         1. c -- has 17 bits
         2. s -- has 16 bits
         3. ovf -- has bit size 1
      4. Interfaces
         1. When doing subtraction, you basically take the second operand and apply 2’s complement and then add that value to the first operand. This component required a lot of interfaces.
            1. Wire m serves to store the result of ANDing two 1s and gets input as the carry in of the first full adder. It is also used as an input of 16 XOR gates to generate 2’s complement of input b.
            2. Wires d, e, f, g, h, i, j, k, l, n, o, p, q, r, t, u each serves as a wire to store the result of XORing wire m and each bit in input b. These are then used as the inputs for input b to the 16 full adders.
            3. Wire v serves to store the result of applying XOR on the most significant bit of input a and wire d which is the first input b into the first full adder. It is then fed into an AND gate as an input to determine if overflow occurred.
            4. Wire w serves to store the result of applying XNOR on the most significant bit of output s and wire d which is the first input b into the first full adder. It is then fed into an AND gate as an input to determine if overflow occurred.
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. This component takes in 2 16-bit inputs and subtracts them and displays the difference as a 16-bit output. It also displays if overflow occurred because of the subtraction.
      7. Documentation: Written by Prathyusha Thiruvuri
   5. Decoder, 4-to-16 bit
      1. Name of Verilog Module: decoder
      2. Inputs
         1. a -- has bit size 4
      3. Outputs
         1. d -- has bit size 16
      4. Interfaces
         1. Four wires are used to have the inverted values of the 4 input bits of a.
            1. nota0 is a wire that stores the negated value of a[0].
            2. nota1 is a wire that stores the negated value of a[1].
            3. nota2 is a wire that stores the negated value of a[2].
            4. nota3 is a wire that stores the negated value of a[3].
         2. These 4 wires are used as inputs to AND gates to determine the values of individual bits in output d.
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. This component serves as a decoder that takes in an n-bit input and returns a 2n- bit one hot output. In this case n is 4. The bit that is high in the output is determined by the value of the input.
      7. Documentation
         1. Written by Prathyusha Thiruvuri
   6. Encoder, 16-to-4 bit
      1. Name of Verilog Module: encoder
      2. Inputs
         1. a -- has bit size 16
      3. Outputs
         1. e -- has bit size 4
      4. Interfaces
         1. We used wires to check if the encoder input was a one hot input
            1. Wires v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15 are used to store if the input is a one-hot.
            2. Wire v stores the result of the above 16 wires ORed together. It is then used in an in-line structural if to see if the result is one-hot.
      5. Any Controls
         1. No controls are used for this component.
      6. Description
         1. An encoder takes in a one-hot input and based on the location of the one-hot bit, returns that index position as a binary number.
      7. Documentation
         1. The input validation to the encoder was derived from the fun.v file provided by Dr. Becker. The rest of the module was written by Prathyusha Thiruvuri.
   7. NAND gate, 16 bit
      1. Name of Verilog Module: nand16
      2. Inputs
         1. in -- has bit size 16
         2. in1 -- has bit size 16
      3. Outputs
         1. out -- has bit size 16
      4. Interfaces
         1. No interfaces are used in this component.
      5. Any Controls
         1. No controls are used in this component.
      6. Description
         1. This module serves to take in 2 binary strings of 16 bits each and does bit-wise NAND on them. It then outputs the result.
      7. Documentation
         1. Written by Rami Jaber
   8. OR gate, 16 bit
      1. Name of Verilog Module: or16
      2. Inputs
         1. in -- has bit size 16
         2. in1 -- has bit size 16
      3. Outputs
         1. out -- has bit size 16
      4. Interfaces
         1. No interfaces are used in this component.
      5. Any Controls
         1. No controls are used in this component.
      6. Description
         1. This module serves to take in 2 binary strings of 16 bits each and does bit-wise OR on them. It then outputs the result.
      7. Documentation
         1. Written by Rami Jaber
   9. NOR gate, 16 bit
      1. Name of Verilog Module: nor16bit
      2. Inputs
         1. A -- has bit size 16
         2. B -- has bit size 16
      3. Outputs
         1. out -- has bit size 16
      4. Interfaces
         1. No interfaces are used in this component.
      5. Any Controls
         1. No controls are used in this component.
      6. Description
         1. This module serves to take in 2 binary strings of 16 bits each and does bit-wise NOR on them. It then outputs the result.
      7. Documentation
         1. Written by Yashal Saleem
   10. XNOR gate, 16 bit
       1. Name of Verilog Module: xnor16bit
       2. Inputs
          1. A -- has bit size 16
          2. B -- has bit size 16
       3. Outputs
          1. out -- has bit size 16
       4. Interfaces
          1. No interfaces are used in this component.
       5. Any Controls
          1. No controls are used in this component.
       6. Description
          1. This module serves to take in 2 binary strings of 16 bits each and does bit-wise XNOR on them. It then outputs the result.
       7. Documentation
          1. Written by Yashal Saleem
   11. Right Arbiter, 16 bit
       1. Name of Verilog Module: rightarbiter
       2. Inputs
          1. A -- has bit size 16
       3. Outputs
          1. out -- has bit size 16
       4. Interfaces
          1. 16-bit wire interface called ‘cas’
       5. Any Controls
          1. No controls are used in this component.
       6. Description
          1. This module serves to take in a binary string of 16 bits and returns a one hot binary string with the one-hot being the LSB that was equal to 1 in the original input.
       7. Documentation
          1. Dally, WIlliam J., Harting, Curtis, R. *Digital Design, A Systems Approach,* Cambridge, Cambridge University Press, 2012
   12. Left Arbiter, 16 bit
       1. Name of Verilog Module: leftarbiter
       2. Inputs
          1. A -- has bit size 16
       3. Outputs
          1. out -- has bit size 16
       4. Interfaces
          1. 16-bit wire interface called ‘cas’
       5. Any Controls
          1. No controls are used in this component.
       6. Description
          1. This module serves to take in a binary string of 16 bits and returns a one hot binary string with the one-hot being the MSB that was equal to 1 in the original input.
       7. Documentation
          1. Dally, WIlliam J., Harting, Curtis, R. *Digital Design, A Systems Approach,* Cambridge, Cambridge University Press, 2012
   13. AND gate, 4 bit
       1. Name of Verilog Module: and4
       2. Inputs
          1. in1 -- has bit size 4
          2. in2 -- has bit size 4
       3. Outputs
          1. out -- has bit size 4
       4. Interfaces
          1. No interfaces are used in this component.
       5. Controls
          1. No controls are used in this component.
       6. Description
          1. This module takes in two binary string of 4 bits and performs bitwise AND on them. The resulting 4 bit string is the output.
       7. Documentation
          1. Written by Khalid Shaik
   14. 16 bit Multiplier
       1. Name of Verilog Module: vedic\_16x16
       2. Inputs
          1. in1Multiplier -- has bit size 16
          2. in2Multiplier-- had bit size 16
       3. Outputs
          1. outMultiplier -- has bit size 32
       4. Interfaces
          1. a,b are the two 16 bit inputs
          2. c is the output from the multiplication
          3. q0,q1,q2,q3 are the 16 bit resultants, from multiplying each part of a and b together
          4. temp1,temp2,temp3,temp4 are the 16 bit intermediary wires that hold the results of the addition of the q resultant wires.
          5. q4 is the 16 bit resultant from q1 and temp1
          6. q5,q6 are 24 bit resultants from the additions of the temp wires
       5. Any Controls
          1. No controls are used for this component.
       6. Description
          1. This is a 16 bit x 16 bit multiplier, using the vedic architecture. It takes in two 16 bits numbers, then multiplies them and outputs the 32 bit resultant.
       7. Documentation: designed off of diagram from <https://pdfs.semanticscholar.org/8387/35eca7bf59a9186824c452fe1476d32cd948.pdf>
   15. Inverter, 16bit
       1. Name of Verilog Module: inv16
       2. Inputs
          1. in -- has bit size 16
       3. Outputs
          1. out -- has bit size 4
       4. Interfaces
          1. This component does not use any interfaces
       5. Any Controls
          1. No controls are used for this component.
       6. Description
          1. A 16 bit inverter, it takes in a 16 bit number, then inverts it out and outputs the result
       7. Documentation
          1. Loosely based off of inverter from ch7 notes provided by Dr. Becker
   16. Multiplexer, 16-to-1 bit
       1. Name of Verilog Module: Multiplexer
       2. Inputs
          1. Ch15,Ch14,Ch13,Ch12,Ch11,Ch10,Ch9,Ch8,Ch7,Ch6,Ch5,Ch4,Ch3,Ch2, Ch1, Ch0, all 16 bit channels
          2. S, the 16 bit selector (one hot)
       3. Outputs
          1. B, the 16 bit output of the mux
       4. Interfaces
          1. This component does not use any interfaces.
       5. Any Controls
          1. The control in this is input s, a one hot 16 bit value to select which channel will be the output
       6. Description
          1. This mux has 16 channels of 16 bit numbers, with a 16 bit one hot selector. The output depends on the channel specified by the selector
       7. Documentation
          1. Based on the mux provided by Dr. Becker in the ch7 verilog notes