GF2 Interim Report 2

Joseph Roberts
jr592

May 29, 2015

1 User guide

1.1 Launching and opening a definition file

1. Run the executable by either typing `./bin/runner` from the project directory, or symlinking the executable into one of the folders on your `PATH` and running it as you would any other program.
2. To bring up the Open File dialog, either press the hotkey `ctrl+o`, or navigate the menu bar at the top to `'File->Open Definition File'`.

1.2 Exploring a definition file

The simulator window is broken into three resizeable panes:

- The top left pane is the Network View. Click the arrow to the left of any subnetwork expand its list of components. Double click on any item to bring it up in the Component View.
- The top right pane is the ComponentView. A list of the inputs and outputs of the curerntly selected component are shown here.
- The bottom pane shows the Plot View. Each monitored signal will generate a trace in this view. You may zoom in and out by scrolling the mouse wheel inside this view, and pan side to side with the scroll bar at the bottom.

1.3 Manipulating monitor points

It should be noted that if a monitor point for a signal is disabled, all record of the trace is lost. Similarly if a monitor point is added part-way through the simulation, the trace for that point will only begin at that point. To add/remove monitor points from the network:

1. Select the component with the signal of interest by double-clicking it in the Network View
2. Select the output(s) which you wish to monitor/unmonitor in the component view by clicking on them
3. Click `'Toggle Monitor Point'` to toggle whether or not the signal is monitored

1.4 Manipulating network inputs (switches)

To change the value of Network inputs:

1. Select the Root Network by double clicking on it in the Network View
2. Select the input(s) that you wish to change by clicking on them in the Component View
3. Click `Switch Input State` to toggle the value of each selected input

2 Example definition files

In this section several example definition files are declared: A 4-bit synchronous counter based on JK flip flops, a full adder, a 4-bit ripple carry adder, and a 4-bit carry lookahead adder. The include functionality of our simulator is demonstrated in both of the 4-bit adders (they include the full adder), as well as a further definition called `4bit-adder-test.def` which includes both the 4-bit adders wired to two synchronous counters as a demonstration.

To demonstrate how easily these definition files might be generated automatically, an example 16-bit carry lookahead adder is also included. This definition is more than 15000 characters long even with all the spaces and newlines removed. Even larger examples are just as trivial to generate, though would be impractical to include in a report. For instance a 64 bit carry lookahead adder requires nearly half a million characters to define - about 100 pages at the same font size as the 16 bit variant.

It should be noted that although the following files are in strict lower case, the format is fully case insensitive such that the requirements in the specification about uppercase items are fufilled. It should also be noted that the ordering of the keys does not matter.

2.1 Full adder

```
{
    inputs : {
        a : false,
        b : false,
        c_in : false
    },
    outputs : {
        s : xor2,
        c_out : or1
    },
    components : {
        xor1 : {
            type : xor,
            inputs : {
                i1 : inputs.a,
                i2 : inputs.b
            }
        },
        xor2 : {
            type : xor,
            inputs : {
                i1 : xor1,
                i2 : inputs.c_in
            }
        },
        and1 : {
            type : and,
            inputs : {
                i1 : xor1,
                i2 : inputs.c_in
            }
        },
        and2 : {
            type : and,
            inputs : {
                i1 : inputs.a,
                i2 : inputs.b
            }
        },
        or1 : {
            type : or,
```

```
            inputs : {
                i1 : and1,
                i2 : and2
            }
        }
    }
}
```

2.2 4 bit synchronous counter

```
{
    inputs: {
        clock: false
    },
    components: {
        jk0: {
            inputs: {
                k: const.true,
                j: const.true,
                clock: inputs.clock
            },
            type: jk
        },
        jk1: {
            inputs: {
                k: and1,
                j: and1,
                clock: inputs.clock
            },
            type: jk
        },
        jk2: {
            inputs: {
                k: and2,
                j: and2,
                clock: inputs.clock
            },
            type: jk
        },
        jk3: {
            inputs: {
                k: and3,
                j: and3,
                clock: inputs.clock
            },
            type: jk
        },
        and1: {
            inputs: {
                i1: jk0.q
            },
            type: and
        },
        and3: {
            inputs: {
                i1: jk0.q,
                i3: jk2.q,
                i2: jk1.q
            },
            type: and
        },
```

```
        and2: {
            inputs: {
                i1: jk0.q,
                i2: jk1.q
            },
            type: and
        }
    },
    outputs: {
        count[4]: {
            1: jk1.q,
            0: jk0.q,
            3: jk3.q,
            2: jk2.q
        }
    }
}
```

2.3 4 bit ripple carry adder

```
{
    inputs: {
        b[4]: false,
        carry: false,
        a[4]: false
    },
    components: {
        adder2: {
            inputs: {
                a: inputs.a[2],
                b: inputs.b[2],
                c_in: adder1.c_out
            },
            type: includes.adder
        },
        adder3: {
            inputs: {
                a: inputs.a[3],
                b: inputs.b[3],
                c_in: adder2.c_out
            },
            type: includes.adder
        },
        adder0: {
            inputs: {
                a: inputs.a[0],
                b: inputs.b[0],
                c_in: inputs.carry
            },
            type: includes.adder
        },
        adder1: {
            inputs: {
                a: inputs.a[1],
                b: inputs.b[1],
                c_in: adder0.c_out
            },
            type: includes.adder
        }
    },
    outputs: {
```

```
        carry : adder3.c_out,
        out[4] : {
            1 : adder1.s,
            0 : adder0.s,
            3 : adder3.s,
            2 : adder2.s
        }
    },
    includes: {
        example_defs/full_adder.def: adder
    }
}
```

2.4 4 bit carry lookahead adder

```
{
  inputs: {
    b[4]: false,
    carry: false,
    a[4]: false
  },
  components: {
    int2-2: {
      inputs: {
        i1: g2
      },
      type: and
    },
    int2-3: {
      inputs: {
        i1: g2,
        i2: p3
      },
      type: and
    },
    g3: {
      inputs: {
        i1: inputs.a[3],
        i2: inputs.b[3]
      },
      type: and
    },
    g2: {
      inputs: {
        i1: inputs.a[2],
        i2: inputs.b[2]
      },
      type: and
    },
    adder0: {
      inputs: {
        a: inputs.a[0],
        b: inputs.b[0],
        c_in: inputs.carry
      },
      type: includes.adder
    },
    adder1: {
      inputs: {
        a: inputs.a[1],
        b: inputs.b[1],
```

```
    c_in: c1
  },
  type: includes.adder
},
int1-1: {
  inputs: {
    i1: g1
  },
  type: and
},
adder2: {
  inputs: {
    a: inputs.a[2],
    b: inputs.b[2],
    c_in: c2
  },
  type: includes.adder
},
adder3: {
  inputs: {
    a: inputs.a[3],
    b: inputs.b[3],
    c_in: c3
  },
  type: includes.adder
},
g1: {
  inputs: {
    i1: inputs.a[1],
    i2: inputs.b[1]
  },
  type: and
},
g0: {
  inputs: {
    i1: inputs.a[0],
    i2: inputs.b[0]
  },
  type: and
},
int1-3: {
  inputs: {
    i1: g1,
    i3: p3,
    i2: p2
  },
  type: and
},
int1-2: {
  inputs: {
    i1: g1,
    i2: p2
  },
  type: and
},
int1: {
  inputs: {
    i1: inputs.carry,
    i3: p1,
    i2: p0
```

```
  },
  type: and
},
int0: {
  inputs: {
    i1: inputs.carry,
    i2: p0
  },
  type: and
},
int3: {
  inputs: {
    i1: inputs.carry,
    i3: p1,
    i2: p0,
    i5: p3,
    i4: p2
  },
  type: and
},
int2: {
  inputs: {
    i1: inputs.carry,
    i3: p1,
    i2: p0,
    i4: p2
  },
  type: and
},
c3: {
  inputs: {
    i1: int2,
    i3: int1-2,
    i2: int0-2,
    i4: int2-2
  },
  type: or
},
c2: {
  inputs: {
    i1: int1,
    i3: int1-1,
    i2: int0-1
  },
  type: or
},
c1: {
  inputs: {
    i1: int0,
    i2: int0-0
  },
  type: or
},
p2: {
  inputs: {
    i1: inputs.a[2],
    i2: inputs.b[2]
  },
  type: xor
},
```

```
  p3: {
    inputs: {
      i1: inputs.a[3],
      i2: inputs.b[3]
    },
    type: xor
  },
  p0: {
    inputs: {
      i1: inputs.a[0],
      i2: inputs.b[0]
    },
    type: xor
  },
  p1: {
    inputs: {
      i1: inputs.a[1],
      i2: inputs.b[1]
    },
    type: xor
  },
  int3-3: {
    inputs: {
      i1: g3
    },
    type: and
  },
  int0-0: {
    inputs: {
      i1: g0
    },
    type: and
  },
  int0-1: {
    inputs: {
      i1: g0,
      i2: p1
    },
    type: and
  },
  int0-2: {
    inputs: {
      i1: g0,
      i3: p2,
      i2: p1
    },
    type: and
  },
  int0-3: {
    inputs: {
      i1: g0,
      i3: p2,
      i2: p1,
      i4: p3
    },
    type: and
  }
},
outputs: {
  carry: adder3.c_out,
```

```
    out[4]: {
      1: adder1.s,
      0: adder0.s,
      3: adder3.s,
      2: adder2.s
    }
  },
  includes: {
    example_defs/full_adder.def: adder
  }
}
```

2.5 4 bit adder test

```
{
    includes : {
        example_defs/4bit-synch-counter.def : counter,
        example_defs/4bit-carry-adder.def : ripple-adder,
        example_defs/4bit-lookahead-adder.def : lookahead-adder
    },
    components : {
        clock : {
            type : siggen,
            config : {
                data : 01,
                period : 10
            }
        },
        counter1 : {
            type : includes.counter,
            inputs : {
                clock : clock
            }
        },
        counter2 : {
            type : includes.counter,
            inputs : {
                clock : counter1.count[3]
            }
        },
        ripple : {
            type : includes.ripple-adder,
            inputs : {
                a[] : counter1.count[],
                b[] : counter2.count[],
                carry : const.false
            }
        },
        lookahead : {
            type : includes.lookahead-adder,
            inputs : {
                a[] : counter1.count[],
                b[] : counter2.count[],
                carry : const.false
            }
        }
    }
}
```

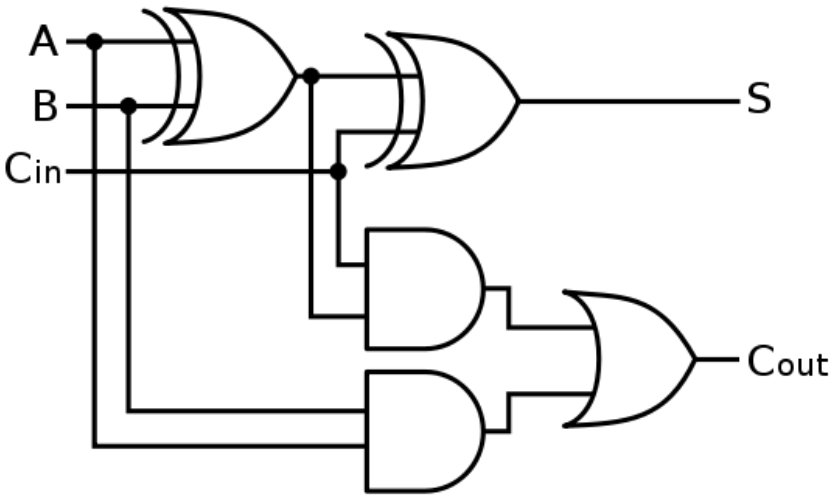

Figure 1: Full adder schematic

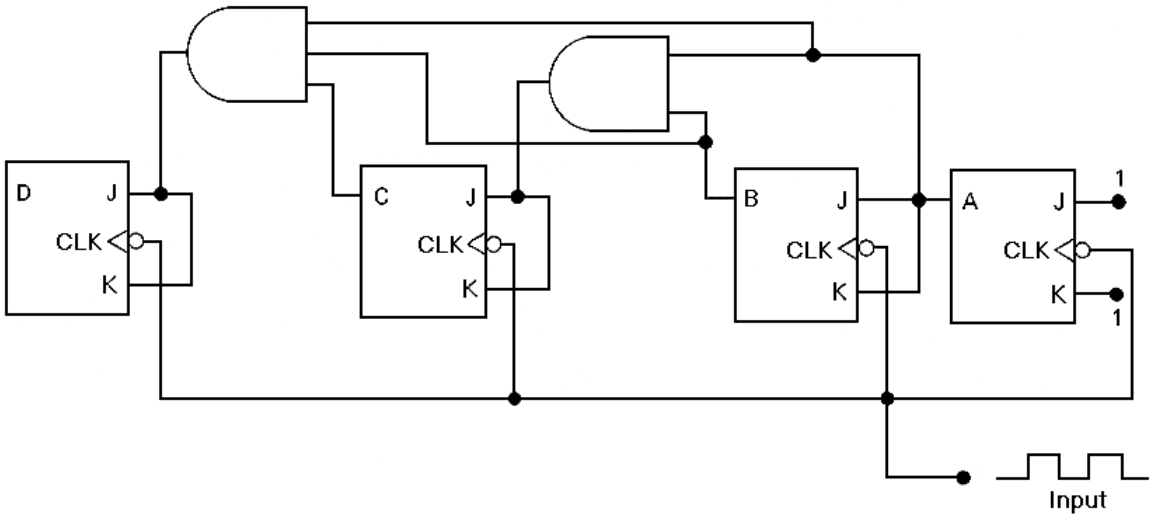

Figure 2: Synchronous counter schematic. Signals marked A, B, C, and D are referenced by the output vector count[4] in the definition

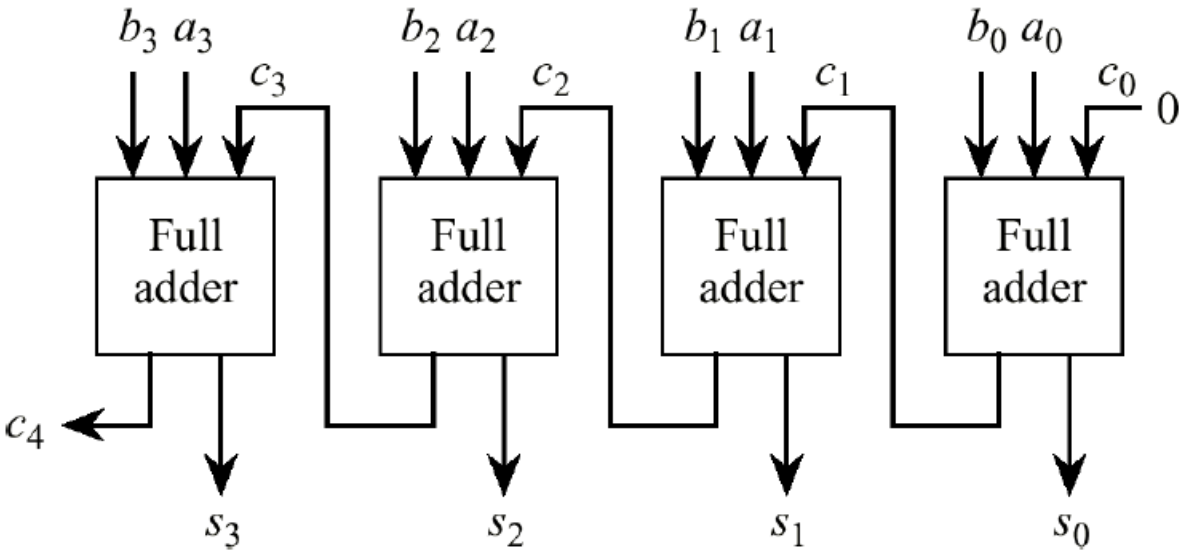

Figure 3: Ripple carry adder schematic

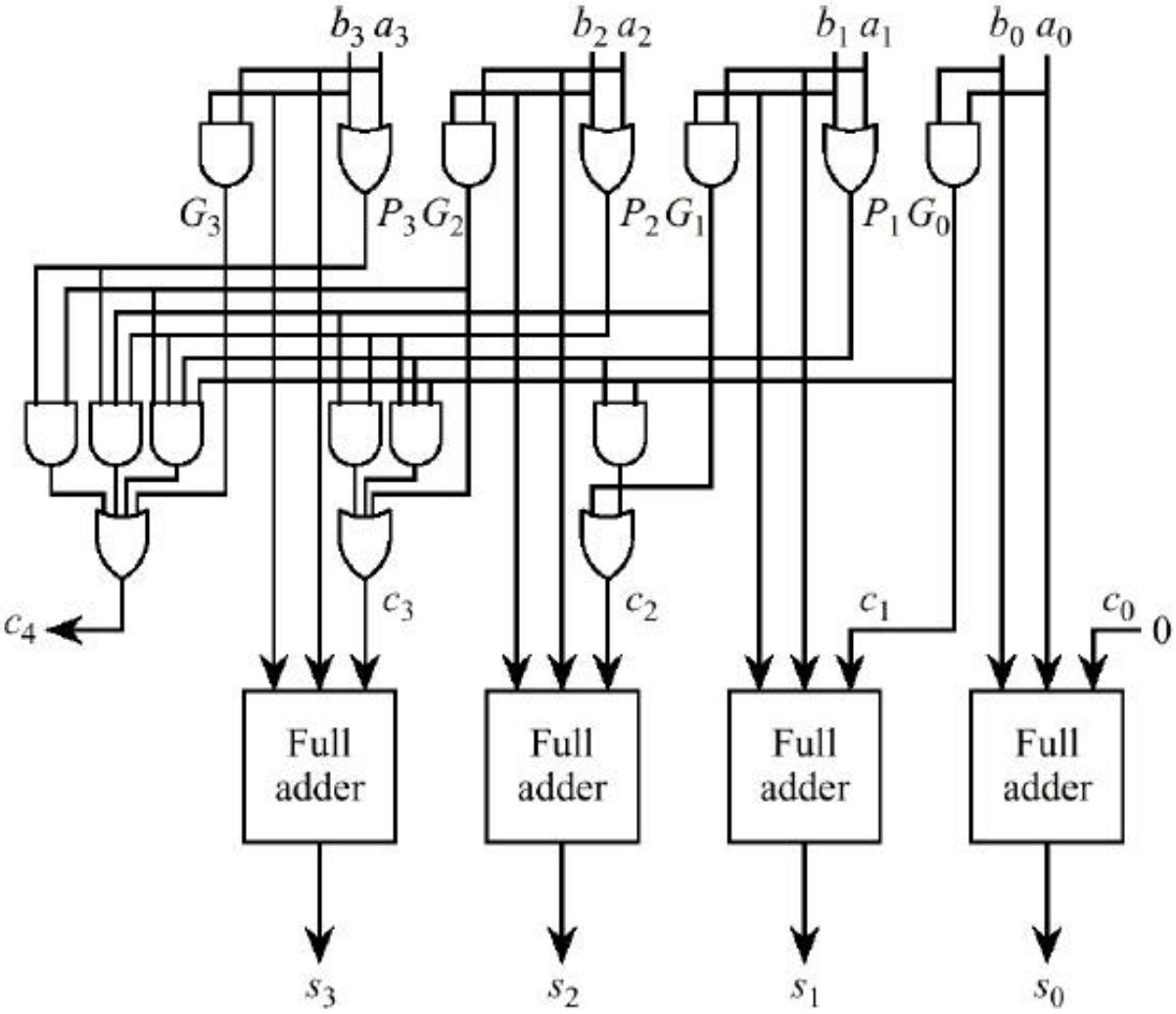

Figure 4: Carry lookahead adder schematic

2.6 16 bit carry lookahead adder

The following definition is stored as a single line with no spaces. Line breaks are inserted here at regular intervals for readability, without regard to whether they intersect a word. Doing so to an actual definition file would prevent it from being parsed.

{inputs:{carry:false,b[16]:false,a[16]:false},components:{int1-15:{inputs:{i9:p9,i8:p8,i1:g1,i11:
p11,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6,i10:p10,i13:p13,i14:p14,i12:p12,i15:p15},type:and},int1-1
4:{inputs:{i9:p9,i8:p8,i1:g1,i11:p11,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6,i10:p10,i13:p13,i14:p14,
i12:p12},type:and},int1-11:{inputs:{i9:p9,i8:p8,i1:g1,i11:p11,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6
,i10:p10},type:and},int1-10:{inputs:{i9:p9,i8:p8,i1:g1,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6,i10:p1
0},type:and},int1-13:{inputs:{i9:p9,i8:p8,i1:g1,i11:p11,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6,i10:p
10,i13:p13,i12:p12},type:and},int1-12:{inputs:{i9:p9,i8:p8,i1:g1,i11:p11,i3:p3,i2:p2,i5:p5,i4:p4,
i7:p7,i6:p6,i10:p10,i12:p12},type:and},c2:{inputs:{i1:int1,i3:int1-1,i2:int0-1},type:or},c7:{inpu
ts:{i8:int6-6,i1:int6,i3:int1-6,i2:int0-6,i5:int3-6,i4:int2-6,i7:int5-6,i6:int4-6},type:or},int12
-12:{inputs:{i1:g12},type:and},int12-13:{inputs:{i1:g12,i2:p13},type:and},int12-14:{inputs:{i1:g1
2,i3:p14,i2:p13},type:and},int12-15:{inputs:{i1:g12,i3:p14,i2:p13,i4:p15},type:and},c9:{inputs:{i
9:int7-8,i8:int6-8,i1:int8,i3:int1-8,i2:int0-8,i5:int3-8,i4:int2-8,i7:int5-8,i6:int4-8,i10:int8-8
},type:or},c6:{inputs:{i1:int5,i3:int1-5,i2:int0-5,i5:int3-5,i4:int2-5,i7:int5-5,i6:int4-5},type:
or},adder14:{inputs:{a:inputs.a[14],b:inputs.b[14],c_in:c14},type:includes.adder},adder15:{inputs
:{a:inputs.a[15],b:inputs.b[15],c_in:c15},type:includes.adder},int5-13:{inputs:{i9:p13,i8:p12,i1:
g5,i3:p7,i2:p6,i5:p9,i4:p8,i7:p11,i6:p10},type:and},int5-12:{inputs:{i8:p12,i1:g5,i3:p7,i2:p6,i5:
p9,i4:p8,i7:p11,i6:p10},type:and},adder10:{inputs:{a:inputs.a[10],b:inputs.b[10],c_in:c10},type:i
ncludes.adder},adder11:{inputs:{a:inputs.a[11],b:inputs.b[11],c_in:c11},type:includes.adder},adde
r12:{inputs:{a:inputs.a[12],b:inputs.b[12],c_in:c12},type:includes.adder},adder13:{inputs:{a:inpu
ts.a[13],b:inputs.b[13],c_in:c13},type:includes.adder},c13:{inputs:{i9:int7-12,i8:int6-12,i1:int1
2,i11:int9-12,i3:int1-12,i2:int0-12,i5:int3-12,i4:int2-12,i7:int5-12,i6:int4-12,i10:int8-12,i13:i
nt11-12,i14:int12-12,i12:int10-12},type:or},c12:{inputs:{i9:int7-11,i8:int6-11,i1:int11,i11:int9-
11,i3:int1-11,i2:int0-11,i5:int3-11,i4:int2-11,i7:int5-11,i6:int4-11,i10:int8-11,i13:int11-11,i12
:int10-11},type:or},c11:{inputs:{i9:int7-10,i8:int6-10,i1:int10,i11:int9-10,i3:int1-10,i2:int0-10
,i5:int3-10,i4:int2-10,i7:int5-10,i6:int4-10,i10:int8-10,i12:int10-10},type:or},c10:{inputs:{i9:i
nt7-9,i8:int6-9,i1:int9,i11:int9-9,i3:int1-9,i2:int0-9,i5:int3-9,i4:int2-9,i7:int5-9,i6:int4-9,i1
0:int8-9},type:or},c15:{inputs:{i9:int7-14,i8:int6-14,i16:int14-14,i1:int14,i11:int9-14,i3:int1-1
4,i2:int0-14,i5:int3-14,i4:int2-14,i7:int5-14,i6:int4-14,i10:int8-14,i13:int11-14,i14:int12-14,i1
2:int10-14,i15:int13-14},type:or},int8-8:{inputs:{i1:g8},type:and},g7:{inputs:{i1:inputs.a[7],i2:
inputs.b[7]},type:and},g6:{inputs:{i1:inputs.a[6],i2:inputs.b[6]},type:and},g5:{inputs:{i1:inputs
.a[5],i2:inputs.b[5]},type:and},g4:{inputs:{i1:inputs.a[4],i2:inputs.b[4]},type:and},g3:{inputs:{
i1:inputs.a[3],i2:inputs.b[3]},type:and},g2:{inputs:{i1:inputs.a[2],i2:inputs.b[2]},type:and},g1:
{inputs:{i1:inputs.a[1],i2:inputs.b[1]},type:and},g0:{inputs:{i1:inputs.a[0],i2:inputs.b[0]},type
:and},int14-14:{inputs:{i1:g14},type:and},int14-15:{inputs:{i1:g14,i2:p15},type:and},g9:{inputs:{
i1:inputs.a[9],i2:inputs.b[9]},type:and},g8:{inputs:{i1:inputs.a[8],i2:inputs.b[8]},type:and},p10
:{inputs:{i1:inputs.a[10],i2:inputs.b[10]},type:xor},p11:{inputs:{i1:inputs.a[11],i2:inputs.b[11]
},type:xor},p12:{inputs:{i1:inputs.a[12],i2:inputs.b[12]},type:xor},p13:{inputs:{i1:inputs.a[13],
i2:inputs.b[13]},type:xor},p14:{inputs:{i1:inputs.a[14],i2:inputs.b[14]},type:xor},p15:{inputs:{i
1:inputs.a[15],i2:inputs.b[15]},type:xor},int4-12:{inputs:{i9:p12,i8:p11,i1:g4,i3:p6,i2:p5,i5:p8,
i4:p7,i7:p10,i6:p9},type:and},int4-13:{inputs:{i9:p12,i8:p11,i1:g4,i3:p6,i2:p5,i5:p8,i4:p7,i7:p10
,i6:p9,i10:p13},type:and},int4-10:{inputs:{i1:g4,i3:p6,i2:p5,i5:p8,i4:p7,i7:p10,i6:p9},type:and},
int4-11:{inputs:{i8:p11,i1:g4,i3:p6,i2:p5,i5:p8,i4:p7,i7:p10,i6:p9},type:and},int4-14:{inputs:{i9
:p12,i8:p11,i1:g4,i11:p14,i3:p6,i2:p5,i5:p8,i4:p7,i7:p10,i6:p9,i10:p13},type:and},int4-15:{inputs
:{i9:p12,i8:p11,i1:g4,i11:p14,i3:p6,i2:p5,i5:p8,i4:p7,i7:p10,i6:p9,i10:p13,i12:p15},type:and},int
9-15:{inputs:{i1:g9,i3:p11,i2:p10,i5:p13,i4:p12,i7:p15,i6:p14},type:and},int9-14:{inputs:{i1:g9,i
3:p11,i2:p10,i5:p13,i4:p12,i6:p14},type:and},int9-11:{inputs:{i1:g9,i3:p11,i2:p10},type:and},int9
-10:{inputs:{i1:g9,i2:p10},type:and},int9-13:{inputs:{i1:g9,i3:p11,i2:p10,i5:p13,i4:p12},type:and
},int9-12:{inputs:{i1:g9,i3:p11,i2:p10,i4:p12},type:and},int8-14:{inputs:{i1:g8,i3:p10,i2:p9,i5:p
12,i4:p11,i7:p14,i6:p13},type:and},int8-15:{inputs:{i8:p15,i1:g8,i3:p10,i2:p9,i5:p12,i4:p11,i7:p1
4,i6:p13},type:and},int8-12:{inputs:{i1:g8,i3:p10,i2:p9,i5:p12,i4:p11},type:and},int8-13:{inputs:
{i1:g8,i3:p10,i2:p9,i5:p12,i4:p11,i6:p13},type:and},int8-10:{inputs:{i1:g8,i3:p10,i2:p9},type:and
},int8-11:{inputs:{i1:g8,i3:p10,i2:p9,i4:p11},type:and},int5-5:{inputs:{i1:g5},type:and},int7-13:
{inputs:{i1:g7,i3:p9,i2:p8,i5:p11,i4:p10,i7:p13,i6:p12},type:and},int2-8:{inputs:{i1:g2,i3:p4,i2:
p3,i5:p6,i4:p5,i7:p8,i6:p7},type:and},int2-9:{inputs:{i8:p9,i1:g2,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i
6:p7},type:and},int7-12:{inputs:{i1:g7,i3:p9,i2:p8,i5:p11,i4:p10,i6:p12},type:and},int2-2:{inputs

:{i1:g2},type:and},int2-3:{inputs:{i1:g2,i2:p3},type:and},int2-6:{inputs:{i1:g2,i3:p4,i2:p3,i5:p6
,i4:p5},type:and},int2-7:{inputs:{i1:g2,i3:p4,i2:p3,i5:p6,i4:p5,i6:p7},type:and},int2-4:{inputs:{
i1:g2,i3:p4,i2:p3},type:and},int2-5:{inputs:{i1:g2,i3:p4,i2:p3,i4:p5},type:and},int10-12:{inputs:
{i1:g10,i3:p12,i2:p11},type:and},int10-13:{inputs:{i1:g10,i3:p12,i2:p11,i4:p13},type:and},int10-1
0:{inputs:{i1:g10},type:and},int10-11:{inputs:{i1:g10,i2:p11},type:and},int10-14:{inputs:{i1:g10,
i3:p12,i2:p11,i5:p14,i4:p13},type:and},int10-15:{inputs:{i1:g10,i3:p12,i2:p11,i5:p14,i4:p13,i6:p1
5},type:and},int7-15:{inputs:{i9:p15,i8:p14,i1:g7,i3:p9,i2:p8,i5:p11,i4:p10,i7:p13,i6:p12},type:a
nd},int7-14:{inputs:{i8:p14,i1:g7,i3:p9,i2:p8,i5:p11,i4:p10,i7:p13,i6:p12},type:and},int0-14:{inp
uts:{i9:p8,i8:p7,i1:g0,i11:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9,i13:p12,i14:p13,i12:p11
,i15:p14},type:and},int0-15:{inputs:{i9:p8,i8:p7,i16:p15,i1:g0,i11:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7
:p6,i6:p5,i10:p9,i13:p12,i14:p13,i12:p11,i15:p14},type:and},int0-12:{inputs:{i9:p8,i8:p7,i1:g0,i1
1:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9,i13:p12,i12:p11},type:and},int0-13:{inputs:{i9:p
8,i8:p7,i1:g0,i11:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9,i13:p12,i14:p13,i12:p11},type:an
d},int0-10:{inputs:{i9:p8,i8:p7,i1:g0,i11:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9},type:an
d},int0-11:{inputs:{i9:p8,i8:p7,i1:g0,i11:p10,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9,i12:p11}
,type:and},int13-15:{inputs:{i1:g13,i3:p15,i2:p14},type:and},int11-11:{inputs:{i1:g11},type:and},
int11-13:{inputs:{i1:g11,i3:p13,i2:p12},type:and},int11-12:{inputs:{i1:g11,i2:p12},type:and},int1
1-15:{inputs:{i1:g11,i3:p13,i2:p12,i5:p15,i4:p14},type:and},int11-14:{inputs:{i1:g11,i3:p13,i2:p1
2,i4:p14},type:and},int2-10:{inputs:{i9:p10,i8:p9,i1:g2,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i6:p7},type
:and},int2-11:{inputs:{i9:p10,i8:p9,i1:g2,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i6:p7,i10:p11},type:and},
int2-12:{inputs:{i9:p10,i8:p9,i1:g2,i11:p12,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i6:p7,i10:p11},type:and
},int2-13:{inputs:{i9:p10,i8:p9,i1:g2,i11:p12,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i6:p7,i10:p11,i12:p13
},type:and},int2-14:{inputs:{i9:p10,i8:p9,i1:g2,i11:p12,i3:p4,i2:p3,i5:p6,i4:p5,i7:p8,i6:p7,i10:p
11,i13:p14,i12:p13},type:and},int2-15:{inputs:{i9:p10,i8:p9,i1:g2,i11:p12,i3:p4,i2:p3,i5:p6,i4:p5
,i7:p8,i6:p7,i10:p11,i13:p14,i14:p15,i12:p13},type:and},int13-14:{inputs:{i1:g13,i2:p14},type:and
},int1-1:{inputs:{i1:g1},type:and},int1-3:{inputs:{i1:g1,i3:p3,i2:p2},type:and},int1-2:{inputs:{i
1:g1,i2:p2},type:and},int1-5:{inputs:{i1:g1,i3:p3,i2:p2,i5:p5,i4:p4},type:and},int1-4:{inputs:{i1
:g1,i3:p3,i2:p2,i4:p4},type:and},int1-7:{inputs:{i1:g1,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6},type:
and},int1-6:{inputs:{i1:g1,i3:p3,i2:p2,i5:p5,i4:p4,i6:p6},type:and},int1-9:{inputs:{i9:p9,i8:p8,i
1:g1,i3:p3,i2:p2,i5:p5,i4:p4,i7:p7,i6:p6},type:and},int1-8:{inputs:{i8:p8,i1:g1,i3:p3,i2:p2,i5:p5
,i4:p4,i7:p7,i6:p6},type:and},int6-10:{inputs:{i1:g6,i3:p8,i2:p7,i5:p10,i4:p9},type:and},int6-11:
{inputs:{i1:g6,i3:p8,i2:p7,i5:p10,i4:p9,i6:p11},type:and},int6-12:{inputs:{i1:g6,i3:p8,i2:p7,i5:p
10,i4:p9,i7:p12,i6:p11},type:and},int6-13:{inputs:{i8:p13,i1:g6,i3:p8,i2:p7,i5:p10,i4:p9,i7:p12,i
6:p11},type:and},int7-9:{inputs:{i1:g7,i3:p9,i2:p8},type:and},int7-8:{inputs:{i1:g7,i2:p8},type:a
nd},int15-15:{inputs:{i1:g15},type:and},int7-7:{inputs:{i1:g7},type:and},int5-7:{inputs:{i1:g5,i3
:p7,i2:p6},type:and},int8-9:{inputs:{i1:g8,i2:p9},type:and},int5-6:{inputs:{i1:g5,i2:p6},type:and
},int5-11:{inputs:{i1:g5,i3:p7,i2:p6,i5:p9,i4:p8,i7:p11,i6:p10},type:and},int6-8:{inputs:{i1:g6,i
3:p8,i2:p7},type:and},int5-10:{inputs:{i1:g5,i3:p7,i2:p6,i5:p9,i4:p8,i6:p10},type:and},int6-9:{in
puts:{i1:g6,i3:p8,i2:p7,i4:p9},type:and},int6-6:{inputs:{i1:g6},type:and},int6-7:{inputs:{i1:g6,i
2:p7},type:and},int5-15:{inputs:{i9:p13,i8:p12,i1:g5,i11:p15,i3:p7,i2:p6,i5:p9,i4:p8,i7:p11,i6:p1
0,i10:p14},type:and},int5-14:{inputs:{i9:p13,i8:p12,i1:g5,i3:p7,i2:p6,i5:p9,i4:p8,i7:p11,i6:p10,i
10:p14},type:and},adder6:{inputs:{a:inputs.a[6],b:inputs.b[6],c_in:c6},type:includes.adder},adder
7:{inputs:{a:inputs.a[7],b:inputs.b[7],c_in:c7},type:includes.adder},adder4:{inputs:{a:inputs.a[4
],b:inputs.b[4],c_in:c4},type:includes.adder},adder5:{inputs:{a:inputs.a[5],b:inputs.b[5],c_in:c5
},type:includes.adder},adder2:{inputs:{a:inputs.a[2],b:inputs.b[2],c_in:c2},type:includes.adder},
adder3:{inputs:{a:inputs.a[3],b:inputs.b[3],c_in:c3},type:includes.adder},adder0:{inputs:{a:input
s.a[0],b:inputs.b[0],c_in:inputs.carry},type:includes.adder},adder1:{inputs:{a:inputs.a[1],b:inpu
ts.b[1],c_in:c1},type:includes.adder},int5-9:{inputs:{i1:g5,i3:p7,i2:p6,i5:p9,i4:p8},type:and},ad
der8:{inputs:{a:inputs.a[8],b:inputs.b[8],c_in:c8},type:includes.adder},adder9:{inputs:{a:inputs.
a[9],b:inputs.b[9],c_in:c9},type:includes.adder},int5:{inputs:{i1:inputs.carry,i3:p1,i2:p0,i5:p3,
i4:p2,i7:p5,i6:p4},type:and},int4:{inputs:{i1:inputs.carry,i3:p1,i2:p0,i5:p3,i4:p2,i6:p4},type:an
d},int7:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4},type:and},int6:
{inputs:{i8:p6,i1:inputs.carry,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4},type:and},int1:{inputs:{i1:in
puts.carry,i3:p1,i2:p0},type:and},int0:{inputs:{i1:inputs.carry,i2:p0},type:and},int3:{inputs:{i1
:inputs.carry,i3:p1,i2:p0,i5:p3,i4:p2},type:and},int2:{inputs:{i1:inputs.carry,i3:p1,i2:p0,i4:p2}
,type:and},c3:{inputs:{i1:int2,i3:int1-2,i2:int0-2,i4:int2-2},type:or},int3-3:{inputs:{i1:g3},typ
e:and},c1:{inputs:{i1:int0,i2:int0-0},type:or},int9:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i11:p9,i
3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8},type:and},int8:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i3
:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8},type:and},c5:{inputs:{i1:int4,i3:int1-4,i2:int0-4,i5:in
t3-4,i4:int2-4,i6:int4-4},type:or},c4:{inputs:{i1:int3,i3:int1-3,i2:int0-3,i5:int3-3,i4:int2-3},t

ype:or},p2:{inputs:{i1:inputs.a[2],i2:inputs.b[2]},type:xor},p3:{inputs:{i1:inputs.a[3],i2:inputs
.b[3]},type:xor},p0:{inputs:{i1:inputs.a[0],i2:inputs.b[0]},type:xor},p1:{inputs:{i1:inputs.a[1],
i2:inputs.b[1]},type:xor},p6:{inputs:{i1:inputs.a[6],i2:inputs.b[6]},type:xor},p7:{inputs:{i1:inp
uts.a[7],i2:inputs.b[7]},type:xor},p4:{inputs:{i1:inputs.a[4],i2:inputs.b[4]},type:xor},p5:{input
s:{i1:inputs.a[5],i2:inputs.b[5]},type:xor},p8:{inputs:{i1:inputs.a[8],i2:inputs.b[8]},type:xor},
p9:{inputs:{i1:inputs.a[9],i2:inputs.b[9]},type:xor},int13-13:{inputs:{i1:g13},type:and},int6-14:
{inputs:{i9:p14,i8:p13,i1:g6,i3:p8,i2:p7,i5:p10,i4:p9,i7:p12,i6:p11},type:and},int6-15:{inputs:{i
9:p14,i8:p13,i1:g6,i3:p8,i2:p7,i5:p10,i4:p9,i7:p12,i6:p11,i10:p15},type:and},c14:{inputs:{i9:int7
-13,i8:int6-13,i1:int13,i11:int9-13,i3:int1-13,i2:int0-13,i5:int3-13,i4:int2-13,i7:int5-13,i6:int
4-13,i10:int8-13,i13:int11-13,i14:int12-13,i12:int10-13,i15:int13-13},type:or},int3-12:{inputs:{i
9:p11,i8:p10,i1:g3,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8,i10:p12},type:and},int5-8:{inputs:{i1:g5,i
3:p7,i2:p6,i4:p8},type:and},int4-4:{inputs:{i1:g4},type:and},int4-5:{inputs:{i1:g4,i2:p5},type:an
d},int4-6:{inputs:{i1:g4,i3:p6,i2:p5},type:and},int4-7:{inputs:{i1:g4,i3:p6,i2:p5,i4:p7},type:and
},int4-8:{inputs:{i1:g4,i3:p6,i2:p5,i5:p8,i4:p7},type:and},int4-9:{inputs:{i1:g4,i3:p6,i2:p5,i5:p
8,i4:p7,i6:p9},type:and},int9-9:{inputs:{i1:g9},type:and},int3-15:{inputs:{i9:p11,i8:p10,i1:g3,i1
1:p13,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8,i10:p12,i13:p15,i12:p14},type:and},int3-14:{inputs:{i9:
p11,i8:p10,i1:g3,i11:p13,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8,i10:p12,i12:p14},type:and},int3-13:{
inputs:{i9:p11,i8:p10,i1:g3,i11:p13,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8,i10:p12},type:and},int3-9
:{inputs:{i1:g3,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8},type:and},int3-11:{inputs:{i9:p11,i8:p10,i1:
g3,i3:p5,i2:p4,i5:p7,i4:p6,i7:p9,i6:p8},type:and},int3-10:{inputs:{i8:p10,i1:g3,i3:p5,i2:p4,i5:p7
,i4:p6,i7:p9,i6:p8},type:and},int3-8:{inputs:{i1:g3,i3:p5,i2:p4,i5:p7,i4:p6,i6:p8},type:and},int1
5:{inputs:{i9:p7,i8:p6,i16:p14,i1:inputs.carry,i11:p9,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8,
i13:p11,i14:p12,i12:p10,i17:p15,i15:p13},type:and},int14:{inputs:{i9:p7,i8:p6,i16:p14,i1:inputs.c
arry,i11:p9,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8,i13:p11,i14:p12,i12:p10,i15:p13},type:and}
,int7-11:{inputs:{i1:g7,i3:p9,i2:p8,i5:p11,i4:p10},type:and},int7-10:{inputs:{i1:g7,i3:p9,i2:p8,i
4:p10},type:and},int11:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i11:p9,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,
i6:p4,i10:p8,i13:p11,i12:p10},type:and},int10:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i11:p9,i3:p1,i
2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8,i12:p10},type:and},int13:{inputs:{i9:p7,i8:p6,i1:inputs.carry
,i11:p9,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8,i13:p11,i14:p12,i12:p10,i15:p13},type:and},int
12:{inputs:{i9:p7,i8:p6,i1:inputs.carry,i11:p9,i3:p1,i2:p0,i5:p3,i4:p2,i7:p5,i6:p4,i10:p8,i13:p11
,i14:p12,i12:p10},type:and},int3-7:{inputs:{i1:g3,i3:p5,i2:p4,i5:p7,i4:p6},type:and},int3-6:{inpu
ts:{i1:g3,i3:p5,i2:p4,i4:p6},type:and},int3-5:{inputs:{i1:g3,i3:p5,i2:p4},type:and},int3-4:{input
s:{i1:g3,i2:p4},type:and},int0-8:{inputs:{i9:p8,i8:p7,i1:g0,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5},
type:and},int0-9:{inputs:{i9:p8,i8:p7,i1:g0,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5,i10:p9},type:and}
,int0-4:{inputs:{i1:g0,i3:p2,i2:p1,i5:p4,i4:p3},type:and},int0-5:{inputs:{i1:g0,i3:p2,i2:p1,i5:p4
,i4:p3,i6:p5},type:and},int0-6:{inputs:{i1:g0,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5},type:and},int0
-7:{inputs:{i8:p7,i1:g0,i3:p2,i2:p1,i5:p4,i4:p3,i7:p6,i6:p5},type:and},int0-0:{inputs:{i1:g0},typ
e:and},int0-1:{inputs:{i1:g0,i2:p1},type:and},int0-2:{inputs:{i1:g0,i3:p2,i2:p1},type:and},int0-3
:{inputs:{i1:g0,i3:p2,i2:p1,i4:p3},type:and},c8:{inputs:{i9:int7-7,i8:int6-7,i1:int7,i3:int1-7,i2
:int0-7,i5:int3-7,i4:int2-7,i7:int5-7,i6:int4-7},type:or},g15:{inputs:{i1:inputs.a[15],i2:inputs.
b[15]},type:and},g14:{inputs:{i1:inputs.a[14],i2:inputs.b[14]},type:and},g13:{inputs:{i1:inputs.a
[13],i2:inputs.b[13]},type:and},g12:{inputs:{i1:inputs.a[12],i2:inputs.b[12]},type:and},g11:{inpu
ts:{i1:inputs.a[11],i2:inputs.b[11]},type:and},g10:{inputs:{i1:inputs.a[10],i2:inputs.b[10]},type
:and}},outputs:{carry:adder15.c_out,out[16]:{11:adder11.s,10:adder10.s,13:adder13.s,12:adder12.s,
15:adder15.s,14:adder14.s,1:adder1.s,0:adder0.s,3:adder3.s,2:adder2.s,5:adder5.s,4:adder4.s,7:add
er7.s,6:adder6.s,9:adder9.s,8:adder8.s}},includes:{example_defs/full_adder.def:adder}}