
This directory contains an example in SpecC language which describes the the SpecC design methodology. We implement the parity checker which calculates the even parity of the unsigned integer number (will be refined into 32-bit vector).

Directory structure:

```
spec/    - specification model for parity checker
arch/    - architecture model for parity checker
comm/    - communication model for parity checker
rtl/     - behavioral RTL model for parity checker
```

Each directory has a makefile to compile the SpecC codes and to generate an executable binary for the simulation. Simply, you can compile them by:

```
make
```

The executable binary file for the simulation is "tb".
You can run it by:

```
./tb
```

```
#
# Makefile: Parity Checker
#

SCC = scrc

SCCOPT = -vv -ww -g

all: tb

tb: tb.sc io.sc parity.sc even.sc ones.sc
    $(SCC) tb $(SCCOPT)

clean:
    -rm -f *~ *.o *.cc *.h
    -rm -f *.si

/*****
* Title: tb.sc
* Author: Dongwan Shin
* Date: 04/20/2001
* Description: testbench for partiy checker
*****/
```

```

import "io";
import "parity";

behavior Main
{
    unsigned int inport, outport;
    event start, done;

    IO U00(inport, outport, start, done);
    parity U01(inport, outport, start, done);

    int main (void)
    {
        par {
            U00.main();
            U01.main();
        }

        return 0;
    }
};

/*****
* Title: parity.sc
* Author: Dongwan Shin
* Date: 04/20/2001
* Description: Specification model for even parity checker
*****/

import "ones";
import "even";

behavior parity(in unsigned int Inport, out unsigned int Outport,
    in event Start, out event Done)
{
    unsigned int data, ocount;
    event istart, idone;

    even U00(Inport, Outport, Start, Done, data, ocount, istart, idone);
    ones U01(data, ocount, istart, idone);

    void main(void) {
        par {
            U00.main();
            U01.main();
        }
    }
};

```

```

/*****
* Title: ones.sc
* Author: Dongwan Shin
* Date: 04/20/2001
* Description: Specification model for one's counter
*****/

```

```

behavior ones(in unsigned int inport, out unsigned int outport,
             in event start, out event done)

```

```

{
    void main(void) {
        unsigned int data;
        unsigned int ocount;
        unsigned int mask;
        unsigned int temp;

        while (1) {
            wait(start);
            data = inport;
            ocount = 0;
            mask = 1;

            while (data != 0) {
                temp = data & mask;
                ocount = ocount + temp;
                data = data >> 1;
            }

            outport = ocount;
            notify(done);
        }
    }
};

```

```

/*****
* Title: even.sc
* Author: Dongwan Shin
* Date: 04/20/2001
* Description: Specification model for even parity checker
*****/

```

```

// #include <stdio.h>

```

```

behavior even(in unsigned int Inport, out unsigned int Outport,
             in event Start, out event Done, out unsigned int data,
             in unsigned int ocount, out event istory, in event idone)

```

```

{
    void main(void) {
        unsigned int mask;

```

```

while (1) {
    wait(Start);

    data = Inport;
    mask = 0x0001;

    notify(istart); // start one's counter

    wait(idone);    // wait for the result of one's counter

    Outport = ocount & mask; // even parity checker

    notify(Done);
}
}
};

/*****
* Title: io.sc
* Author: Dongwan Shin
* Date: 04/20/2001
* Description: testbench
*****/

#include <stdio.h>
#include <stdlib.h>

// get unsigned integer from stdin
behavior IO(out unsigned int inport, in unsigned int outport,
           out event start, in event done)
{
    void main(void) {
        char buf[16];

        while (1) {
            printf("Input for parity checker: ");
            gets(buf);
            inport = atoi(buf);
            notify(start); // start parity checker

            wait(done);    // check result of parity checker
            printf("parity checker output(%s) = %u\n", buf, outport);
        }
    }
};

```