**Homework – 3**

Hai-Ming, Hsu 433004106

**(10 points) Consider the signal control example on page 2-5 of Lecture 4. Please rewrite its SystemC code using dynamic processes for signals::sig\_ctrl() and signals::red\_sig()**

**Solution:**

**#define SC\_INCLUDE\_DYNAMIC\_PROCESSES**

**class signals : public sc\_module {**

**public:**

**SC\_HAS\_PROCESS(signals);**

**signals(sc\_module\_name nm):sc\_module(nm){**

**SC\_METHOD(run);**

**}**

**void run(){**

**sc\_spawn( sc\_bind(&signals::sig\_ctrl, this));**

**}**

**void sig\_ctrl(){**

**enum directions {RED=‘R’, OFF=‘F’}; char dir;**

**bool did\_red = false;**

**while(true){ cin >> dir;**

**switch(dir){**

**case RED:**

**sc\_process\_handle h = sc\_spawn( sc\_bind(&signals::red\_sig, this, true));**

**wait(h.terminated\_event());**

**did\_red = true;**

**break;**

**case OFF:**

**sc\_process\_handle h = sc\_spawn( sc\_bind(&signals::red\_sig, this, false));**

**wait(h.terminated\_event());**

**did\_red = false;**

**break;**

**}**

**}**

**}**

**void red\_sig(bool on){**

**if(on){**

**wait(3, SC\_MS);**

**cout << "Red light is turned on!" << endl;**

**}**

**else {**

**wait(10, SC\_MS);**

**cout << "Red light is turned off." << endl;**

**}**

**}**

**}**