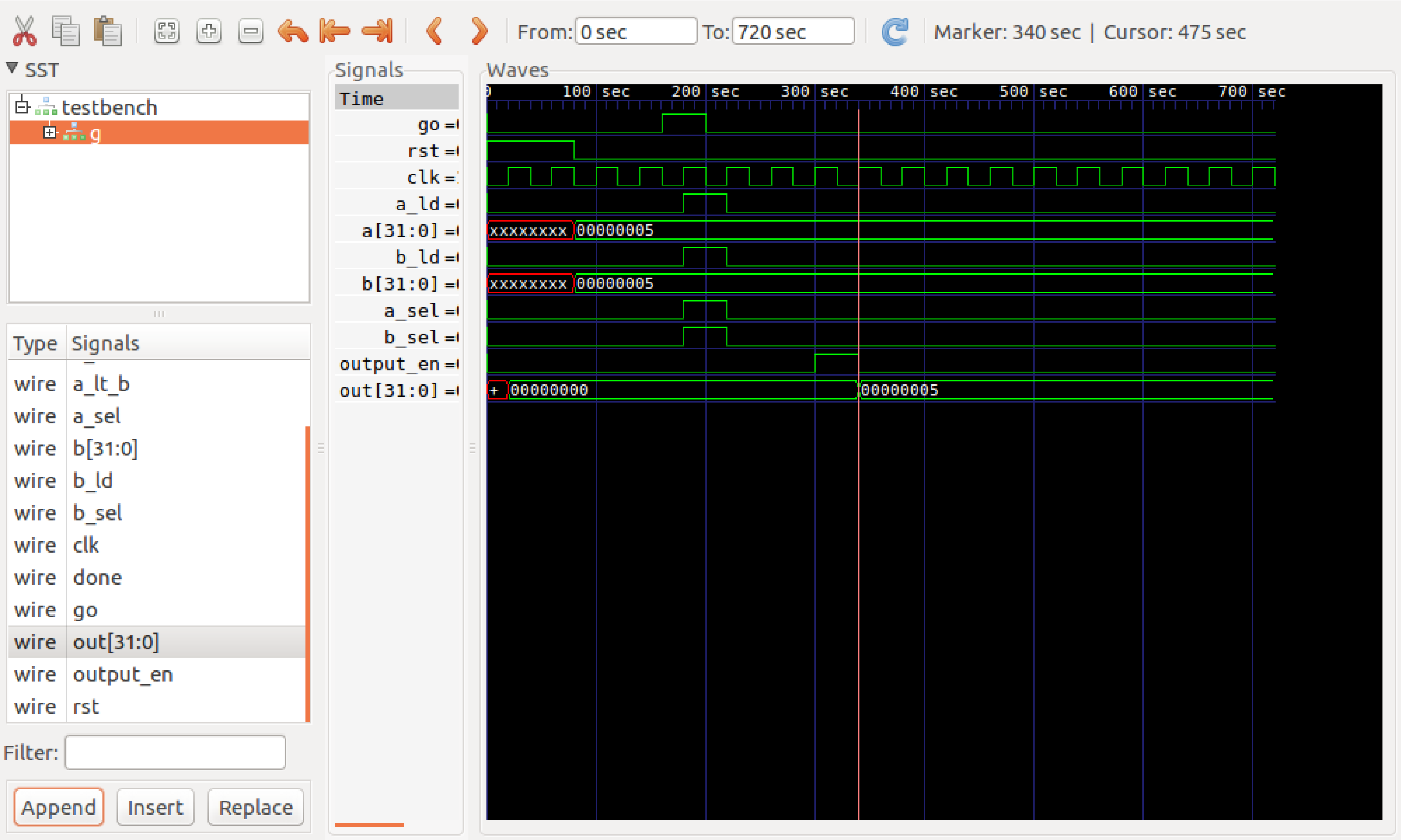
EHD_Report for GCD

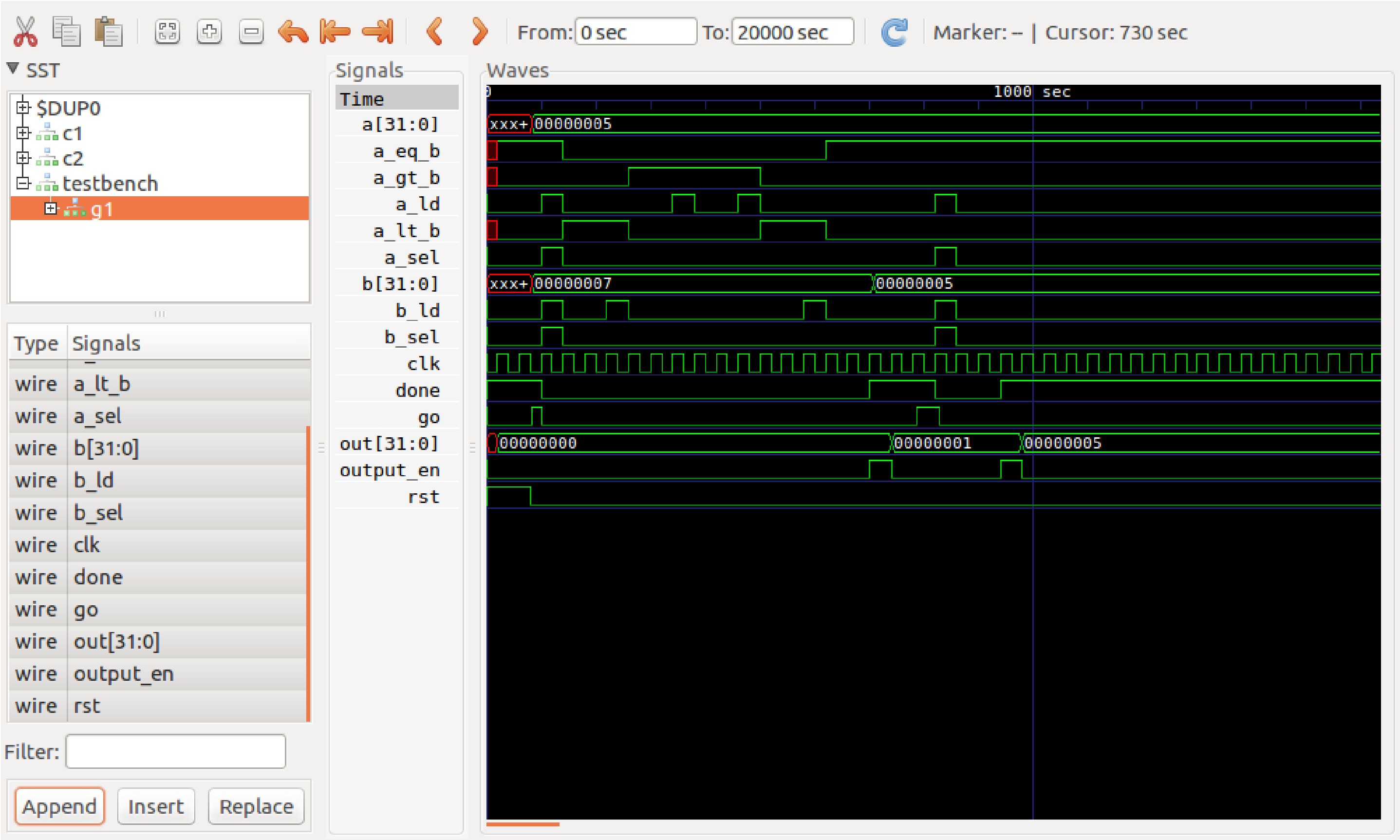
Shivani Chepuri
Roll: 2018122004
UG2 – ECD



From: 0 sec To: 720 sec Marker: 340 sec | Cursor: 475 sec

Type	Signals
wire	a_lt_b
wire	a_sel
wire	b[31:0]
wire	b_ld
wire	b_sel
wire	clk
wire	done
wire	go
wire	out[31:0]
wire	output_en
wire	rst

Filter:



From: 0 sec To: 20000 sec Marker: -- | Cursor: 730 sec

SST

- $DUP0
- c1
- c2
- testbench
  - g1

Type	Signals
wire	a_lt_b
wire	a_sel
wire	b[31:0]
wire	b_ld
wire	b_sel
wire	clk
wire	done
wire	go
wire	out[31:0]
wire	output_en
wire	rst

Filter:

here, gcd(5,7) = 1

Input numbers: a[31:0] (here a = 5), b[31:0](here b = 5)
output: out[31:0] (here, gcd = 5)

go (input): if go is high it means that the state machine is turned on (initially keeping 0, then making 1 for one clock cycle and then 0 again)

rst (input): the circuit works only when this reset is low (i.e, reset is considered to be active high)

clk (input): output changes only at rising posedge of clk

a_ld and b_ld(inputs for datapath and outputs for control) : a register can be read/written only when load is activated
a_sel and b_sel(inputs for datapath and outputs for control): selecting lines for the muxes of respective registers a and b

done: a flag variable to indicate that the gcd is computed

output_en : enable line for the output register [31:0]out. It should be high when the out register’s needs to be altered and then back to low after the value is altered.

The outputs are shown in hex format.

Explanation:

When rst = 1(high / active high), no inputs are given.The machine has to be resetted before enabling

When rst is made zero, the inputs are given to a and b registers
Now, we make go = 1 which enables the machine.

Then, we check if no output is being written to [31:0]out reg. i.e, done = 0. We wait until that state. Once (done == 0) we make go = 0 and then,the gcd_mach starts working.

We must then set output_en = 1 and done = 1 simlutaneously and write the value is out register.

The load lines and select lines are on and off according to the different states they switch into, by the logic

```
    if(a>b) a = a – b; - case 1
    if(a<b) b = b – a; - case 2
```

For instance, in case 1, we need to write a – b value to a reg. So, we select a – b line of the a_mux and make a_load high.