Case Study of Register-Transfer-Level (RTL) design
Shift-Add Multiplier

Madhav Desai

March 20, 2017

Specification

- Input **start**, single bit, asserted by environment to indicate beginning of multiplication.
- Output **done**, single bit, asserted by system to indicate end of multiplication.
- Inputs **A**, **B**, each of 32-bit width, the two numbers to be multiplied.
- Output **result**, 32-bit product (bottom 32-bits of 64 bit true product).
- Inputs **clock**, **reset**: reset is active high, positive edge of **clock** is used.
- System computes the product of **A**, **B**.

RTL Algorithm

```
register AREG[31:0], BREG[31:0]
register PROD[63:0], TSUM[32:0], COUNT[5:0]
thread ShiftAddMul {
 rst: if (start)  then
        COUNT := 33, AREG := A, BREG := B, PROD := 0,
        goto decr
      else goto rst end if
 sumstate:
    COUNT := COUNT - 1,
    TSUM := (PROD[63:32] + (AREG[0] ? BREG : 0)),
    goto update
 update:
    PROD := (TSUM && PROD[31:1]),  AREG := (AREG >> 1)
    if(COUNT == 0) RESULT := PROD[31:0], goto done
    else goto sumstate
 donestate:
    done = 1, goto rst
}
```

The data-path: transfers

Use the following coding to mark the transfers:

```
T0      COUNT := 33
T1      COUNT := COUNT - 1
T2      AREG  := A
T3      AREG  := (AREG >> 1)
T4      BREG  := B
T5      PROD  := 0
T6      PROD  := (0 && TSUM && PROD[32:1])
T7      TSUM  := (PROD[63:32] + (AREG[0]? B : 0)
T8      RESULT := PROD[31:0]
```

The data-path: status signals

Use the following coding for the status predicates:

```
S          (COUNT == 0)
```

The data-path: implement using muxes, combinational circuits, registers

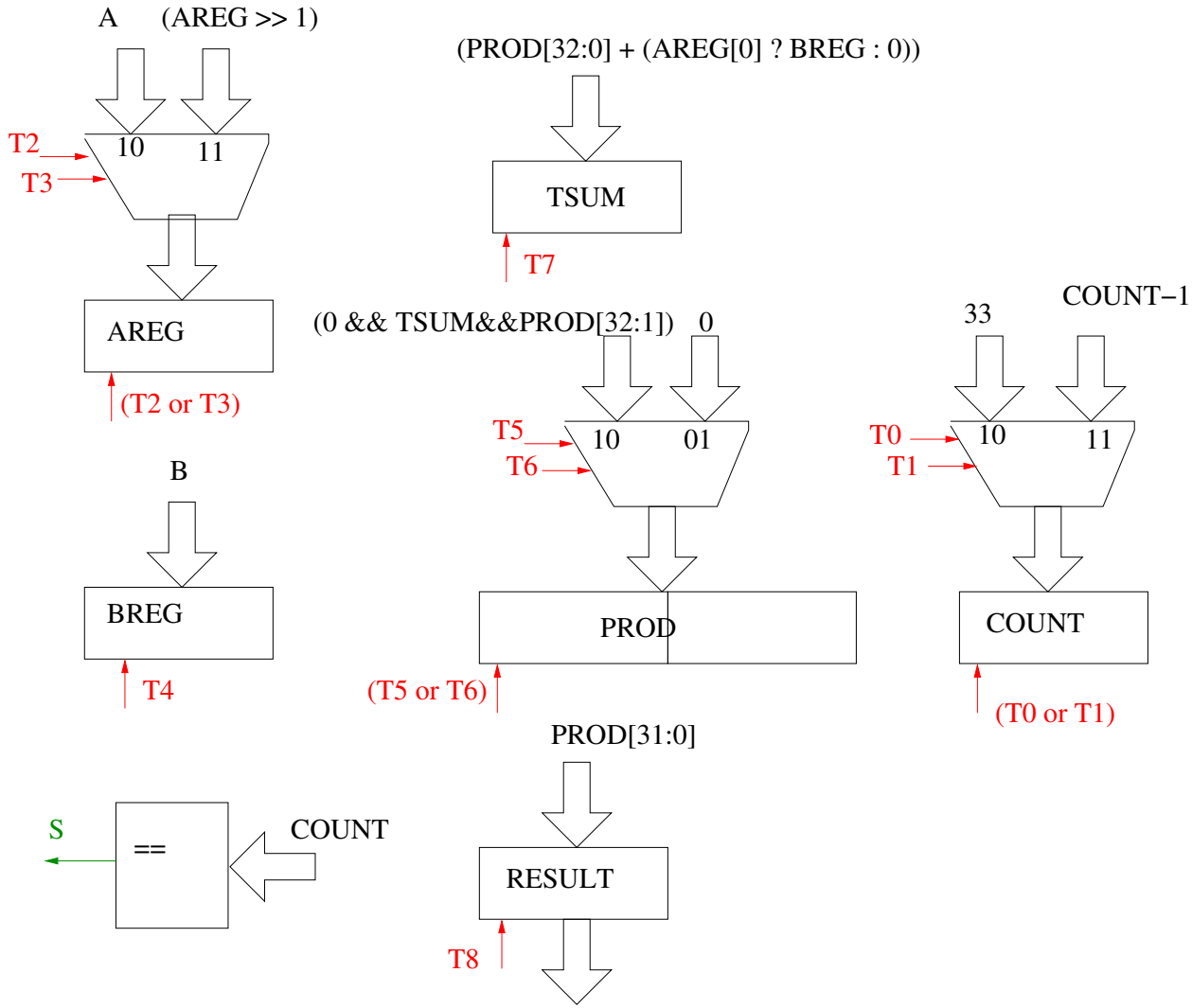

Figure : Datapath for the counter

The control-path: FSM

state	reset	start	S	next-state	T0	T1	T2	T3	T4	T5	T6	T7	T8	done
_	1	_	_	rst	_	_	_	_	_	_	_	_	_	_
rst	0	0	_	rst	0	0	0	0	0	0	0	0	0	0
rst	0	1	_	sumstate	1	0	1	0	1	1	0	0	0	0
sumstate	0	_	_	update	0	1	0	0	0	0	0	1	0	0
update	0	_	0	sumstate	0	0	0	1	0	0	1	0	0	0
update	0	_	1	donestate	0	0	0	1	0	0	1	0	1	0
donestate	0	_	_	rst	0	0	0	0	0	0	0	0	0	1

VHDL

- Entity/Architecture for Datapath.
- Entity/Architecture for Controlpath.
- Entity/Architecture for ShiftAddMultiplier.
- Component package: component declarations for ShiftAddMultiplier, Datapath, Controlpath.
- Entity/Architecture for Testbench.