Draw Block Code Review 1

Michael Li - ml1811@ic.ac.uk

Code review on draw block. Code generated by jbh111 Jake Humphrey, code reviewed by ml1811 Michael Li of Group 36. Code was submitted well before deadline.

1 Synthesis

1.1 Errors

There are no synthesis errors.

1.2 Warnings

```
@W:CG296 : db.vhd(91) | Incomplete sensitivity list - assuming completeness
@W:CG290 : db.vhd(118) | Referenced variable state is not in sensitivity list
@W:CD604 : db.vhd(177) | OTHERS clause is not synthesized
@W:CD638 : db.vhd(31) | Signal pen_x is undriven
@W:CD638 : db.vhd(31) | Signal pen_y is undriven
@W:CD638 : db.vhd(32) | Signal previous_command is undriven
@W:CD638 : draw_any_octant.vhd(40) | Signal x03 is undriven
@W:CD638 : draw_any_octant.vhd(40) | Signal yo3 is undriven
@W:CL240 : db.vhd(29) | dao_xbias is not assigned a value (floating) -- simulation
    mismatch possible.
@W:CL169 : db.vhd(52) | Pruning register prev_command.op(1 downto 0)
@W:CL169 : db.vhd(52) | Pruning register prev_command.pen(1 downto 0)
@W:CL167 : db.vhd(63) | Input xbias of instance dao is floating
@W:CL159 : db.vhd(11) | Input reset is unused
@W:CL159 : db.vhd(21) | Input dbb_rcbclear is unused
```

db.vhd(91) — Incomplete sensitivity list - assuming completeness
Action: TO CHANGE
Reason:
Whilst not mandatory, it is best practice to always put all inputs of the combinatorial process in the sensitivity list. Errors due to omitting signals from sensitivity list in combinatorial processes are very difficult to identify so it is safe to ensure all are explicitly declared.

db.vhd(118) — Referenced variable state is not in sensitivity list
Action: TO CHANGE
Reason:
This combinatorial processes output depends on signal `state` however this process is not asked to trigger when it gets changed, so it won't be updated and potentially give the wrong output.

db.vhd(177) — OTHERS clause is not synthesized
Action: TO CHANGE
Reason:
This others clauses isn't needed as the states are an enumerated type and all possible values are defined so there is no others. Leaving this in is a matter of style as synthesis will trim it anyway. I recommend we remove it.

db.vhd(31) — Signal pen_x is undriven
Action: TO CHANGE
Reason:
It is an unused signal that was not removed from the declarations. We might as well just delete the signal.

db.vhd(31) — Signal pen_y is undriven
Action: TO CHANGE
Reason:
It is an unused signal that was not removed from the declarations. We might as well just delete the signal.

db.vhd(32) — Signal previous_command is undriven
Action: TO CHANGE
Reason:
It is an unused signal that was not removed from the declarations. We might as well just delete the signal. Most likely because it was of the wrong type anyway and you declared a new signal `prev_command` but forgot to remove the old one. The unused declaraed signal should be removed.

draw_any_octant.vhd(40) — Signal x03 is undriven
Action: TO CHANGE
Reason:
It is an unused signal that was not removed from the declarations. The unused declaraed signal should be removed.

draw_any_octant.vhd(40) — Signal yo3 is undriven
Action: TO CHANGE
Reason:
It is an unused signal that was not removed from the declarations. The unused declaraed signal should be removed.

db.vhd(29) — dao_xbias is not assigned a value (floating) – simulation mismatch possible
Action: TO CHANGE
Reason:
Our glorious leader Tom Clarke has yet to divulge the secrets of xbias. Until he does we are to ignore this but the signal should not float. Assign it to a constant.

db.vhd(52) — Pruning register prev_command.op(1 downto 0)
Action: Allowed
Reason:
It was easier to just store the entire previous command, but only previous x and previous y was used. It would be more optimal space wise to only register previous x and y but not necessary as synthesis is optimising it out for us anyway.

db.vhd(52) — Pruning register prev_command.pen(1 downto 0)
Action: Allowed
Reason:
Same as above.

db.vhd(63) — Input xbias of instance dao is floating
Action: TO CHANGE
Reason:
We are not using xbias yet but the signal should be assigned to some constant for now, floating signals are very bad.

db.vhd(11) — Input reset is unused
Action: TO CHANGE
Reason:
It would appear the reset signal is never used in the file. The draw block cant reset, and therefore we cant intialise it. This is a very serious issue that requires immediate addressing. There is no way to reset any of the processes inside db.
db.vhd(21) — Input dbb_rcbclear is unused
Action: Allowed
Reason:
The ram control block will eventually implement the clear screen command so we can just ignore this. There may need to be some logic that forwards this onto the vdp top level entity.

1.3 Summary

Most of the warnings are harmless superfluous signal declarations that can easily be trimmed in the code. There is a rather serious point where the reset of the entire block is not implemented and a potentially dangerous set of floating signals.

2 FSM Operation

FSM reset state. *ref: Line34 in architecture of rtl of db.*
The reset state is the first state listed in the enum type. This violates the specification. Please make another dummy state for simulation to initialise to.

FSM operation This seems to look good apart from the lack of block-wide reset.

3 Reset

The reset has been omitted entirely; this needs to be implemented all over the entire block.

4 Code correctness

Whilst the use of records is nice from a programming perspective as it ties up loose signals, this may cause an issue with post-synthesis vhdl and running that back through the testbench without some kind of explicit adapter. All IF statments in combinatorial proceses have an else or default assignments. There are no combinatorial loops that I can see, there are no multiple driven signals, there are no latches, there are no unsynthesisable constructs such as wait fors or processes with both negative and positive edge triggers. All for loops or while loops have constant limits because there are no loops. There are no latches. There are no uses of real numbers anywhere or gated or multiple clocks. There are no flip flops clocked on by signals. There is no asynchronous set or reset (or any reset present at all). There is nothing before wait until statements in processes. We ensured we compiled under vhdl 93. There was an issue with the shared data bus as I thought DB would not change the shared bus unless I explicitly asked for another command.

5 Design correctness

Design appears sound. No tristated errors from synplify suggest singly driven signals. RCB assumed that drawblock didn't alter data on the bus if busy, however I think that RCB will send the busy signal one cycle too late and it would be better for RCB to register the all of dbbbus.

6 Code style

Good use of generics in the defining of vector sizes. Division of processes seems sane. Lines are not overly long but would be nice if keywords were capitalised. Processes are named consistently and sensibly. An optimisation that could be made, if the line requested to be drawn, the FSM could simply go directly to sending the draw command, as only one pixel is required instead of going through the draw line algorithm.