ScalaHDL

by Yao Li

Outline

- async block
- HDL conversion
- simulation
- future plan

original plan

```
def compare(a: HDLType, b: HDLType, x: HDLType, y: HDLType, dir: Int) {
  if (dir == ASC) {
    when (a > b) {
      x := b
      y := a
    } .otherwise {
      x := a
      y := b
    }
  } else {
    when (a > b) {
      x := a
      y := b
    } .otherwise {
      x := b
      y := a
    }
  }
}
```

but what if...

```
def compare(a: HDLType, b: HDLType, x: HDLType, y: HDLType, dir: Int) {
  if (dir == ASC) {
    x := a
    y := b
    when (a > b) {
      x := b
      y := a
    }
  } else {
    x := b
    x := a
    when (a > b) {
      x := a
      y := b
    }
  }
}
```

async block

```
def compare(a: HDLType, b: HDLType, x: HDLType, y: HDLType, dir: Int) {
  async {
    if (dir == ASC) {
      when (a > b) {
        x := b
        y := a
      } .otherwise {
        x := a
        y := b
      }
    } else {
      when (a > b) {
        x := a
        y := b
      } .otherwise {
        x := b
        y := a
      }
    }
  }
}
```

explicitly declare a sub circuit

HDL conversion

- wire / register
  - a variable is wire if it's in a simple combinational logic sub circuit
  - a combinational logic circuit is simple if it only contains assignments
- input / output
  - a variable is output if it's been assigned

HDL conversion example

```
module FlipFlop (

d,

q,

clk

);



input clk;

input d;

output q;

reg q;



always @(posedge clk) begin: _logic

q <= d;

end




endmodule
```

simulation

- The simulation for FlipFlop has been fixed.

- The simulation for BitonicSort remains incorrect.
  Not fatal. Must be some unnoticed mistakes.

Future Plan

- Fix the BitonicSort simulation.
- Test Bench.

Any Question?

Thanks!