ScalaHDL

by Yao Li

Outline

- Current state of ScalaHDL
  - example: binary to gray conversion
  - example: addition between signed and unsigned registers
- Future plan

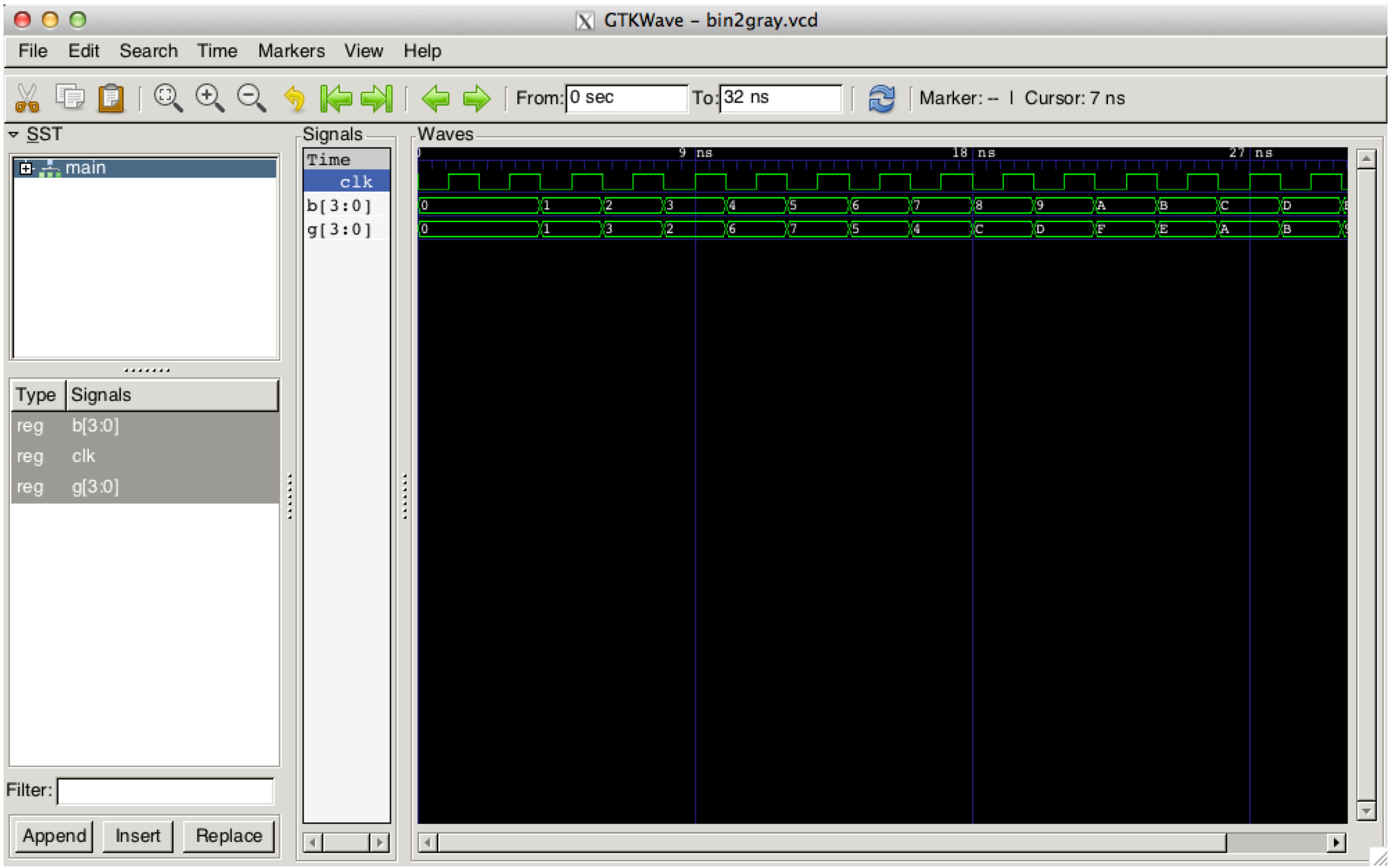
Example:
Binary to Gray Conversion

```
trait Bin2Gray extends ScalaHDL {

  val width = 5

  defMod.bin2gray('g, 'b) {
    async {
      for (i <- 0 until width)
        'g(i) := 'b(i + 1) ^ 'b(i)
      'g(width) := 'b(width)
    }
  }
}
```

Example:
Binary to Gray Conversion

```
module bin2gray (g, b);


input [5:0] b;

output [5:0] g;

wire [5:0] g;


assign g[0] = (b[1] ^ b[0]);

assign g[1] = (b[2] ^ b[1]);

assign g[2] = (b[3] ^ b[2]);

assign g[3] = (b[4] ^ b[3]);

assign g[4] = (b[5] ^ b[4]);

assign g[5] = b[5];


endmodule
```

Example:
Binary to Gray Conversion



Example: Addition between Signed and Unsigned Registers

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    sync(1).add {

      when ('rst is 1) {

        'z := 0

      } .otherwise {

        'z := 'a + 'b

      }

    }

  }

}
```

Example: Addition between Signed and Unsigned Registers

```
module adder (clk, rst, a, b, z);


input [3:0] b;

input clk;

input rst;

input signed [4:0] a;

output signed [5:0] z;

reg signed [5:0] z;


always @(posedge clk) begin: _add

if (rst == 1) begin

z <= 0;

end

else begin

z <= (a + $signed({1'b0, b}));

end

end


endmodule
```

Others

- Edit and add more tests on signals.
  - Fixed some bugs.
- Concat operator is changed from “~~” to “++”.

Future Plan

- ROM, RAM.
- “yield” in test bench.
- Real “if” statement.

Any Question?

Thanks!