**Motivation & Status**

The purpose of the TMRG toolset is to automatize the process of triplicating digital circuits.

The tool:
- is compatible with ASIC design flow used in the HEP (Verilog-95 RTL, Cadence tools)
- does not constrain user's coding style (the source Verilog should be synthesizable)
- allows to obtain various flavors of TMR (registers only, full triplication, logic triplication, clock triplication)
- assists in the physical implementation stage (synthesis, P&R)
- assists designer in the verification process (generation of SEE)
- can be run in a batch mode (fully automatic flow)

Project started: March 2015
- Project size: >13000 lines of code
- Documentation size: 60 pages (pdf)

---

**Constraining the design**

- designer decides which blocks and signals are to be triplicated by using TMRG directives (placed in Verilog code, configuration file, or command line argument)
  - `// tmrg default triplicate_in_var_triplicate`
  - `// tmrg default do_not_triplicate_combLogic`

TMRG automates “conversion” between triplicated and not triplicated signals:
- if non triplicated signal is connected to a triplicated signal a passive fanout is added
- if a triplicated signal is connected to a non triplicated signal a majority voter is added

---

**Tripling Finite State Machine**

```verilog
tmrGroupA # .combLogic() ;
```

---

**SEU / SET Generator (seeg)**

- generates Single Event Effects stimulus to be used for transient simulations

---

**Placement Generator (plag)**

- generates placement directives (for Encounter)

---

**TMRG Toolset**

- `tmrg` - triplicates the Verilog code and generates synthesis constraints (for the Design Compiler)
- `tbg` - generates generic test bench template (with /without TMR, SEE injection, post synthesis, post PNR)
- `plag` - generates placement directives (for Encounter)
- `seeg` - generates Single Event Effects stimulus to be used for transient simulations

---

**TMRG Digital Design Flow**

- **User constraints**
  - functional verification
  - synthesis
  - logic synthesis
  - P&R constraints
  - SEE stimulus
- **Constraining the design**
  - user constraints
  - testbench generation
  - timing verification
  - process
  - tmrg

---

**Triple Modular Redundancy Generator**

http://cern.ch/tmrg

---

**TMRG Process**

- **Behavior Description**
  - Verilog code
  - Logic synthesis
  - Timing verification
- **User Constraints**
  - Functional verification
  - Logic synthesis
  - P&R constraints
  - SEE stimulus
- **Constraining the design**
  - User constraints
  - Testbench generation
  - Timing verification

---

**Motivation & Status**

- The tool is designed to automate the process of triplicating digital circuits.
- It provides various options for triplication (registers only, full triplication, logic triplication, clock triplication).
- It assists in the physical implementation stage (synthesis, P&R) and in the verification process (generation of SEE).
- It can be run in a batch mode for fully automatic flow.

---

**Constraining the design**

- The designer decides which blocks and signals are to be triplicated by using TMRG directives.
- The tool automates the conversion between triplicated and not triplicated signals.
  - Non triplicated signals are connected to triplicated signals with passive fanouts.
  - Triplicated signals are connected to non triplicated signals with majority voters.

---

**Tripling Finite State Machine**

```verilog
module combLogic();
  input a, b, c,
  output out,
  wire in,
  reg state;
  always @ (posedge clk)
    state <= stateNext;
  always @ (posedge clk)
    stateNext <= in ^ state;
  endmodule
```

---

**SEU / SET Generator (seeg)**

- Generates Single Event Effects stimulus to be used for transient simulations.

---

**Placement Generator (plag)**

- Generates placement directives for Encounter.

---

**TMRG Toolset**

- **tmrg** - triplicates the Verilog code and generates synthesis constraints for the Design Compiler.
- **tbg** - generates generic test bench template (with or without TMR, SEE injection, post synthesis, post PNR).
- **plag** - generates placement directives for Encounter.
- **seeg** - generates Single Event Effects stimulus to be used for transient simulations.

---

**TMRG Digital Design Flow**

- User constraints:
  - Functional verification
  - Logic synthesis
  - P&R constraints
  - SEE stimulus
- Constraining the design:
  - User constraints
  - Testbench generation
  - Timing verification
- Process:
  - tmrg

---

**TMRG Process**

- **Behavior Description**:
  - Verilog code
  - Logic synthesis
  - Timing verification
- **User Constraints**:
  - Functional verification
  - Logic synthesis
  - P&R constraints
  - SEE stimulus
- **Constraining the design**:
  - User constraints
  - Testbench generation
  - Timing verification

---

**Motivation & Status**

- The tool aims to automate the process of triplicating digital circuits.
- It supports various triplication options (registers only, full triplication, logic triplication, clock triplication).
- It assists in physical implementation (synthesis, P&R) and verification (SEE generation).
- It can be run in batch mode for automatic processing.

---

**Constraining the design**

- The designer specifies which blocks and signals are to be triplicated using TMRG directives.
- The tool automates the conversion between triplicated and not triplicated signals.
  - Non triplicated signals are connected to triplicated signals with passive fanouts.
  - Triplicated signals are connected to non triplicated signals with majority voters.

---

**Tripling Finite State Machine**

```verilog
module combLogic();
  input a, b, c,
  output out,
  wire in,
  reg state;
  always @ (posedge clk)
    state <= stateNext;
  always @ (posedge clk)
    stateNext <= in ^ state;
  endmodule
```

---

**SEU / SET Generator (seeg)**

- Generates Single Event Effects stimulus.

---

**Placement Generator (plag)**

- Generates placement directives for Encounter.

---

**TMRG Toolset**

- **tmrg** - triplicates the Verilog code and generates synthesis constraints for the Design Compiler.
- **tbg** - generates generic test bench template (with or without TMR, SEE injection, post synthesis, post PNR).
- **plag** - generates placement directives for Encounter.
- **seeg** - generates Single Event Effects stimulus to be used for transient simulations.