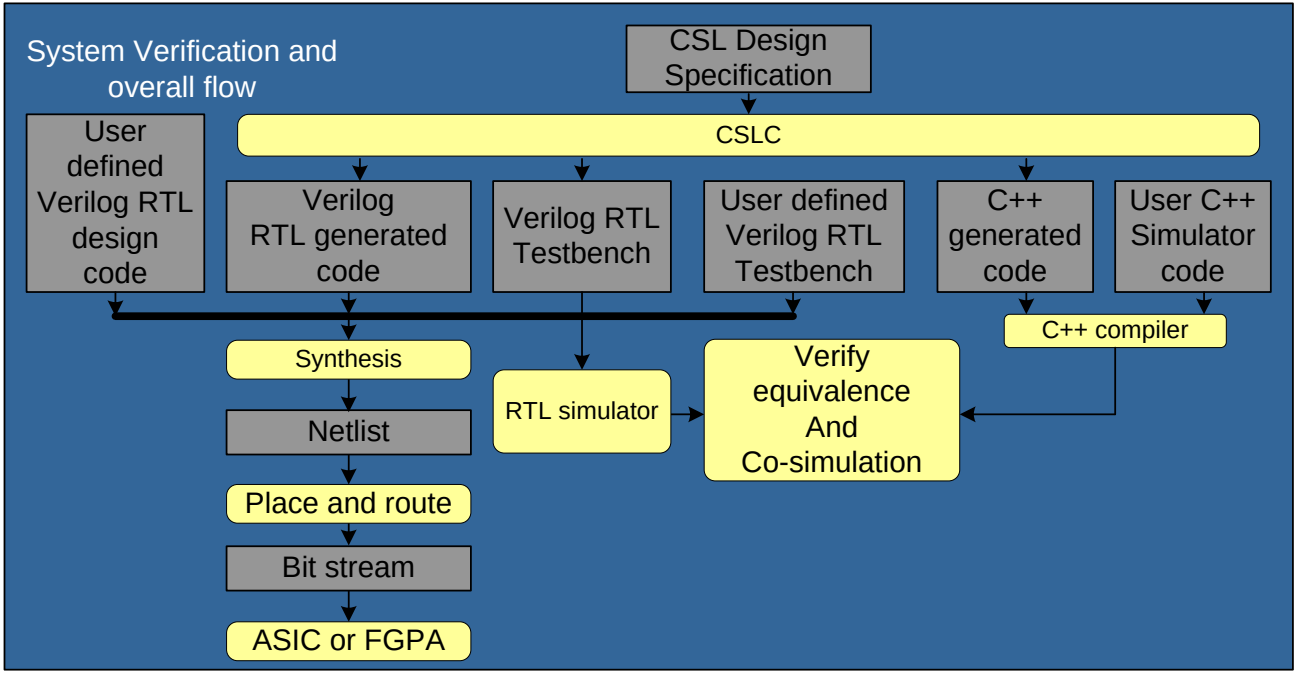

System Verification and overall flow
CSL Design Specification
CSLC
User defined Verilog RTL design code
Verilog RTL generated code
Verilog RTL Testbench
User defined Verilog RTL Testbench
C++ generated code
User C++ Simulator code
C++ compiler
Synthesis
Netlist
Place and route
Bit stream
ASIC or FGPA
RTL simulator
Verify equivalence And Co-simulation

commonly coded modules and reduces the consistency bugs due to interface or constant value differences.

CSL as an abstract modeling language, compliments the Electronic System Level design (ESL) process. By using a high level language with the expressiveness of standard RTL and the power of a second generation RTL language, The CSL language is object oriented. The CSLC tool meets your design team's needs in many ways. Examples include reduced time to a completed design interconnect, reduced time to C/C++ an RTL test, reduced consistency bugs due to interface or constant value differences, and reduced bugs due to bugs in commonly coded modules..
automatically generate module with correct interfaces
automatically generate common components
automatically generate a memory map with no address collisions
automatically generate a memory map connected to state elements (registers and registers file and SRAMs)

Figure 1CSLC inputs and outputs

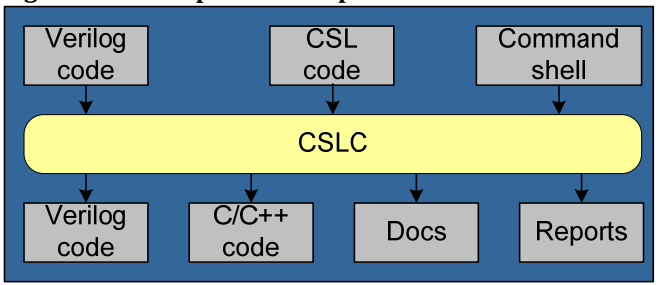
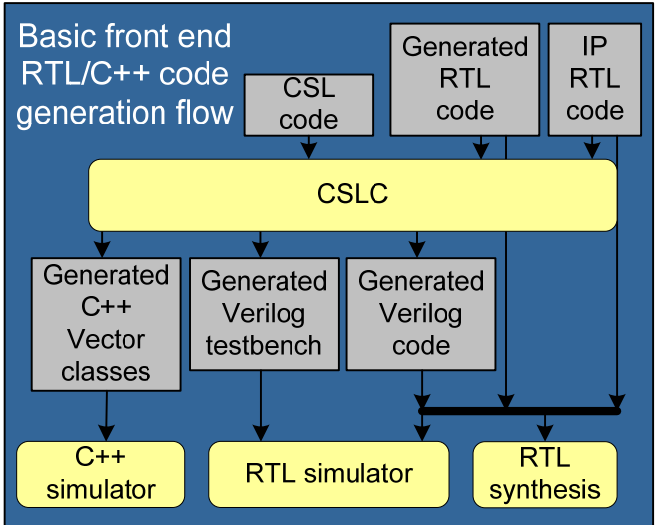


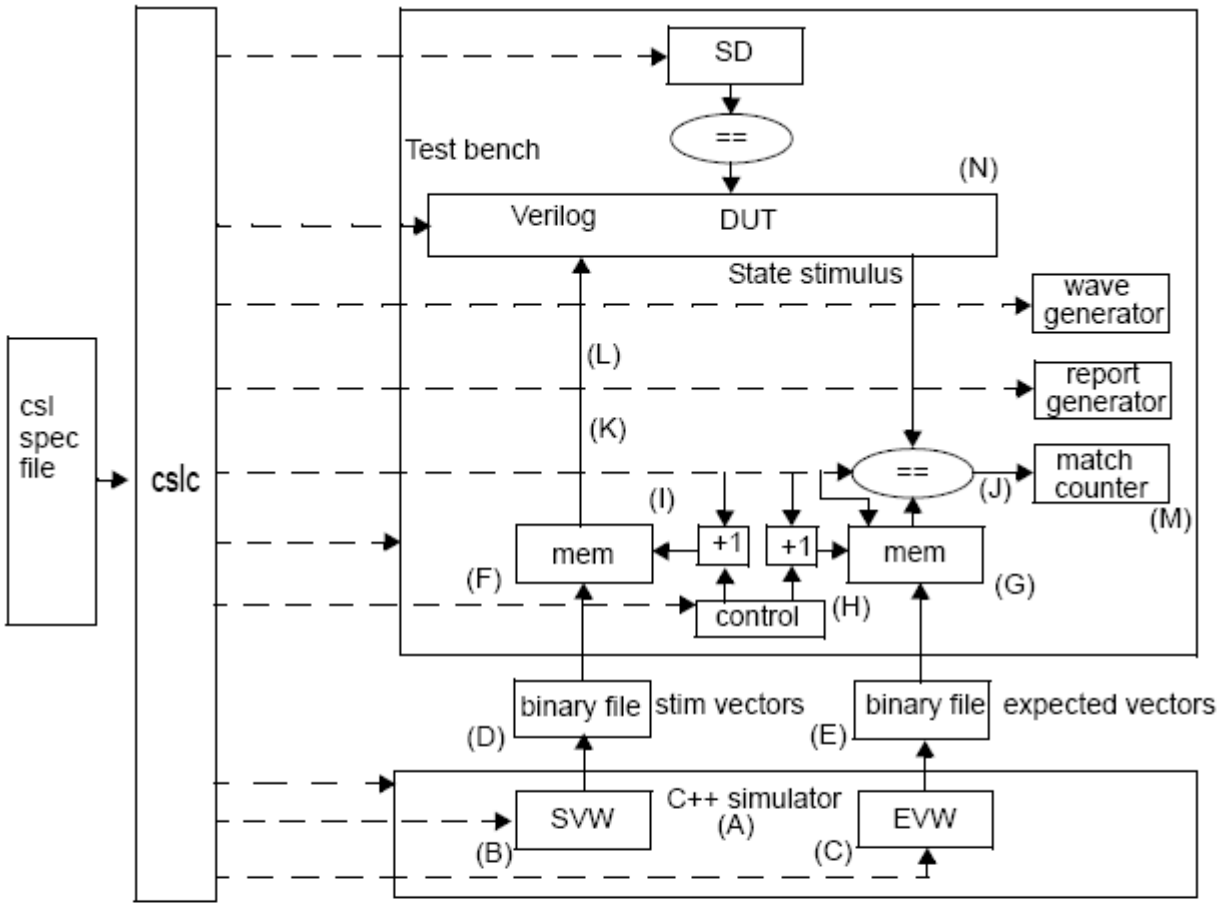
Figure 2 Front end RTL/C++ code generation flow



associates the state element with a architectural test state name, the event used to capture the state, clock domain, reset signal, max number of frames (snap shots of the state element at various times. State is generated by C/C++ simulators. State is used for verification purposes by test benches.

Test bench Generation

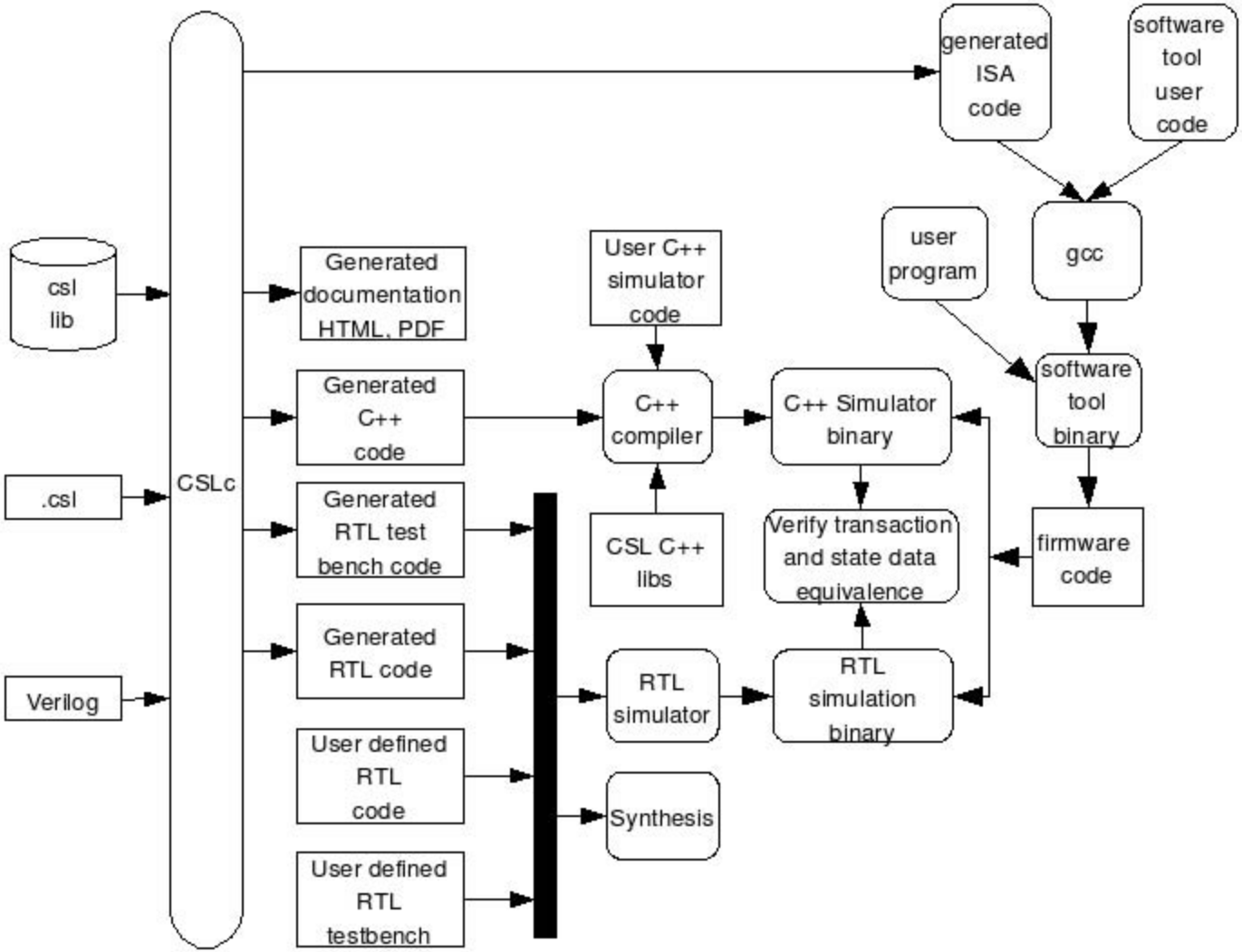
The CSLC tool will automatically generate test benches for each module in the design. The design under test can be driven with test vectors which either loaded written to a file and into a



EVW - expected vector writer
SVW - stimulus vector writer
– — – the component is generated from the CSLC
binary files - can be dumped as state data to save space and speed up the comparison process

A - C++ simulator
B - stimulus vector writer
C - expected vector writer
D - simulus vectors
E - expected vectors
F,G - memory
H - control unit
I - incrementor
J - comparator
K - report generator
L - wave generator
M - match counter
N - DUT (Design Under Test)

Interface consistency


generated ISA code
software tool user code
csl lib
CSLc
Generated documentation HTML, PDF
User C++ simulator code
user program
gcc
.csl
Generated C++ code
C++ compiler
C++ Simulator binary
software tool binary
Generated RTL test bench code
CSL C++ libs
Verify transaction and state data equivalence
firmware code
Verilog
Generated RTL code
RTL simulator
RTL simulation binary
User defined RTL code
Synthesis
User defined RTL testbench