Notes on Interrupt functionality testing

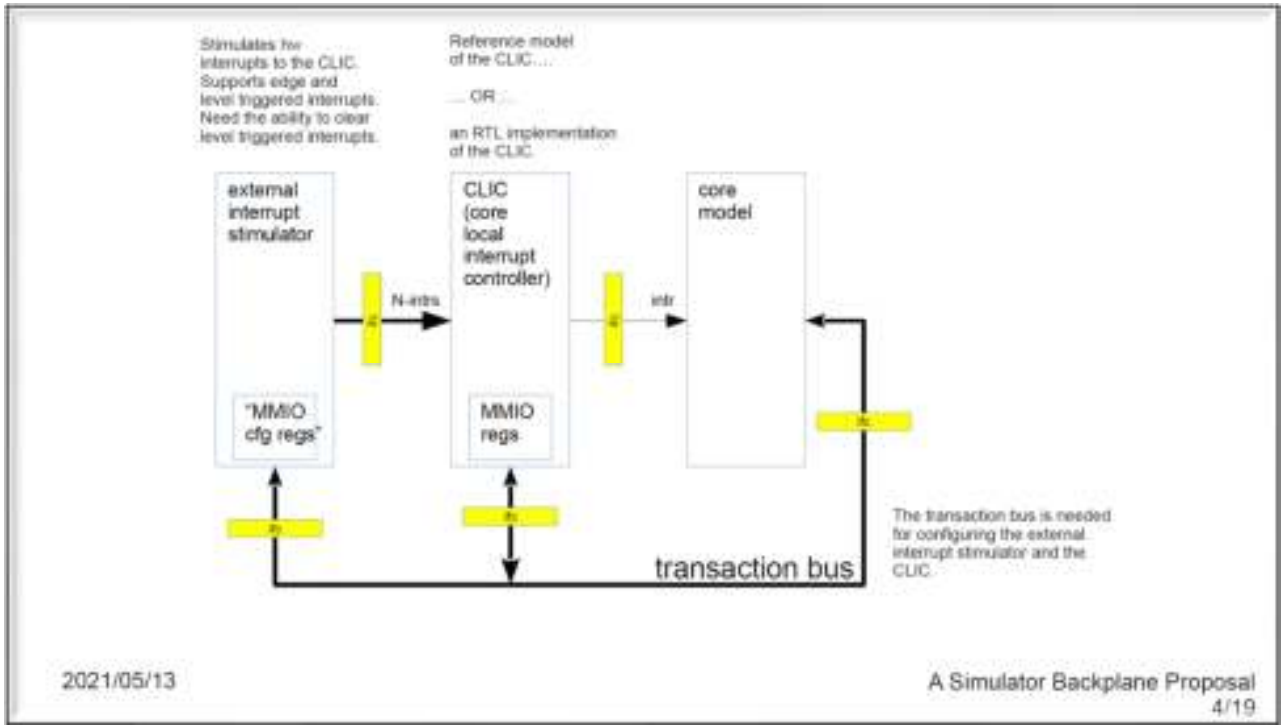
Simon Davidmann, simond@imperas.com

24 June 2021

The issues

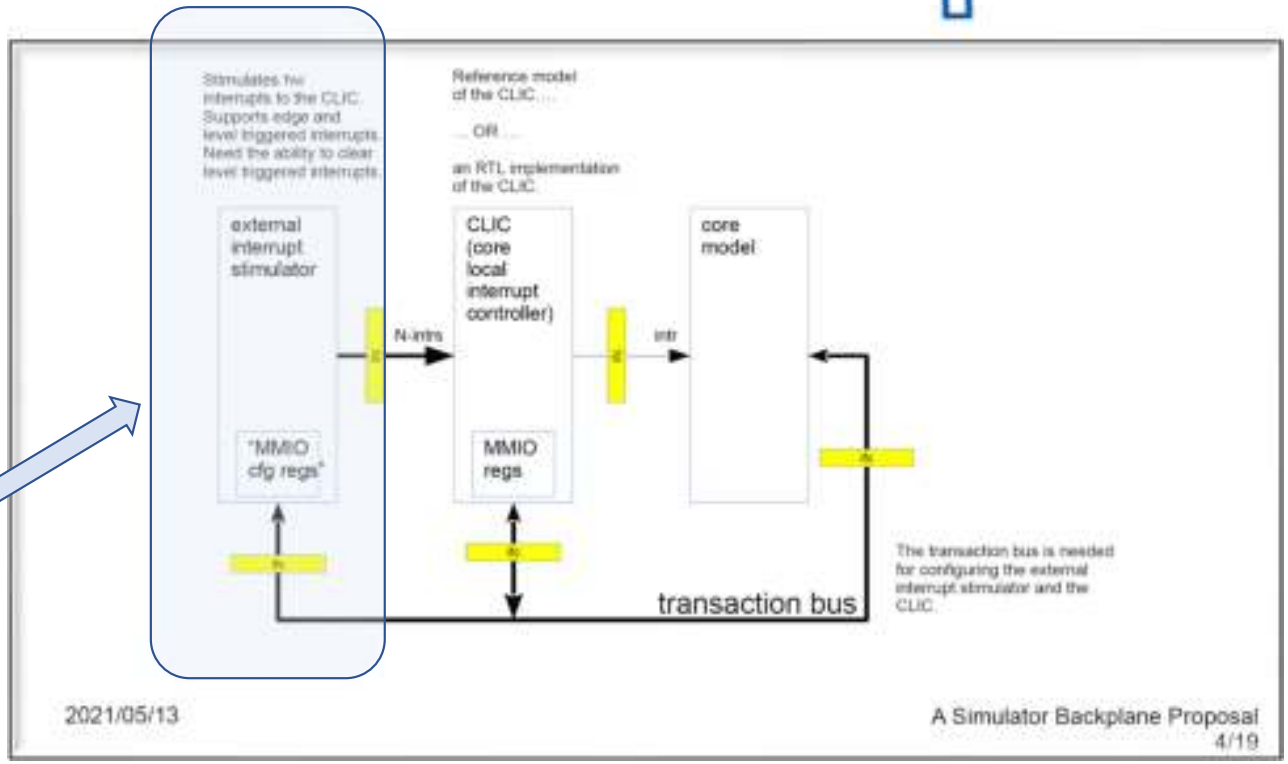
- How to test async events/interrupts for a RISCV CPU
- Requires a testbench/platform that triggers CPU pins…
- Needs software handler to respond to events
- Need to decide on level of ‘verification’
  - e.g. very simple: ‘did the event occur’
  - Or could be much more complex for CPU micro-architectural DV
- Need to define how the tests are written to work with RTL and reference model

Recap – Seagate/Bill's slide…



- Presented to RVI Arch Tests WG – 13/May/2021

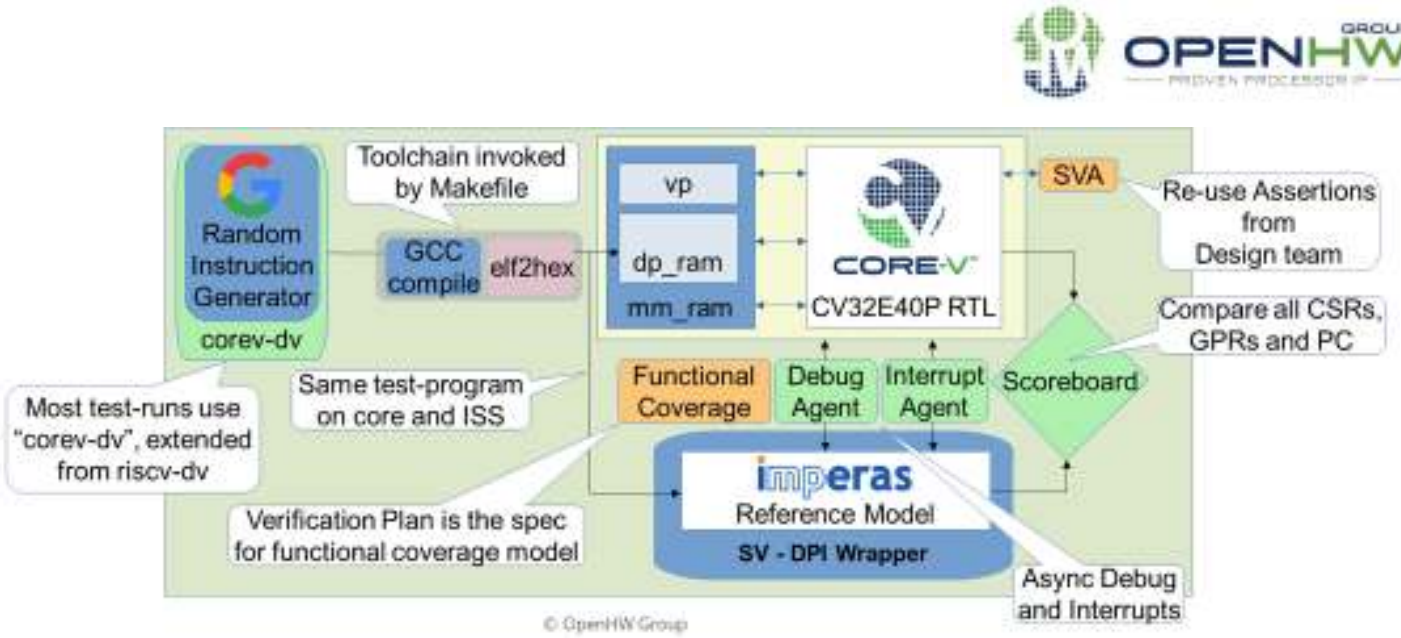
Recap – Seagate/Bill's slide…



Relevant to this discussion

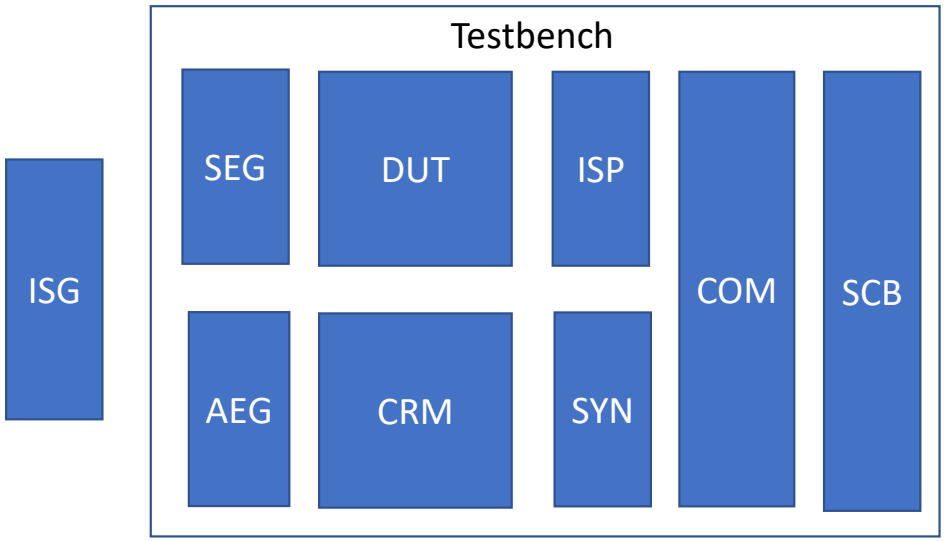
- Presented to RVI Arch Tests WG – 13/May/2021

Example of RISCV CPU Async HW DV flow (lock-step-compare)



Current CV32E40P OpenHW flow
(Imperas model encapsulated in SystemVerilog)

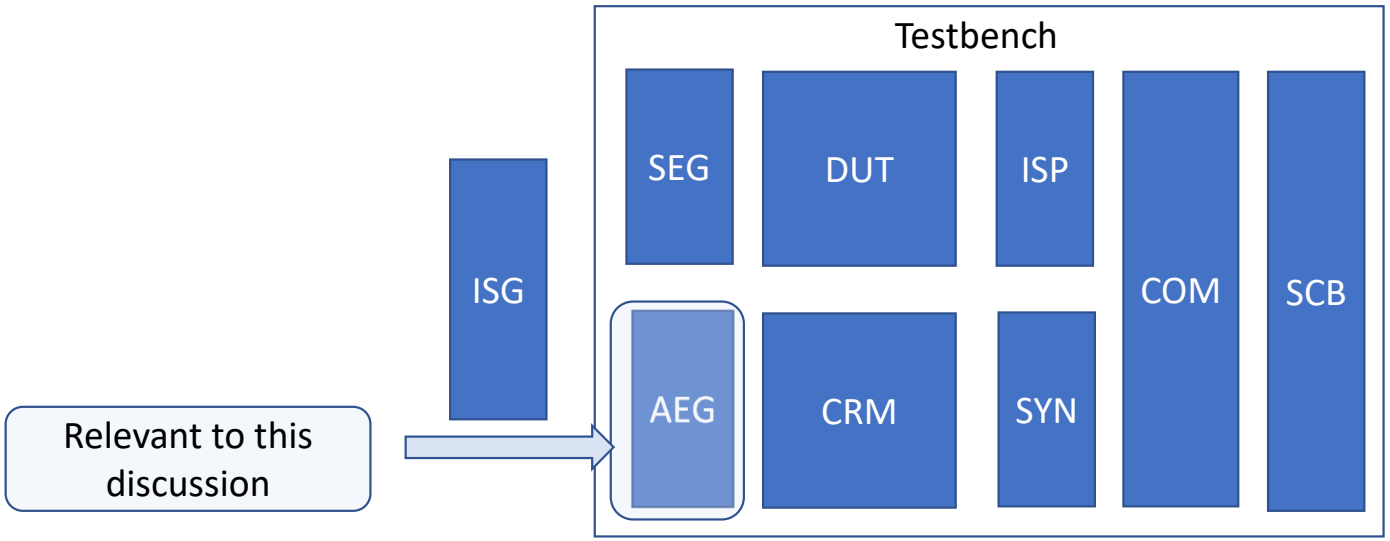
Imperas view:
Components in Async Test Bench (ATB)



- ISG – Instruction Stream Generator
- SEG – Synchronous Event Generator
- AEG – Asynchronous Event Generator
- DUT –the RTL of the RISC-V CPU Device Under Test
- CRM – the Controllable Reference Model (e.g. Imperas configurable envelope model)
- ISP – IntroSPector (tracer) to provide details of what happens in the micro-architecture of the DUT
- SYN - SYNchronizer to keep CRM in sync with DUT
- COM - COMparator between DUT and CRM
- SCB – SCoreBoard to track what has been 'seen' (instructions/events/modes)

=> Complete CPU HW DV environment for best results

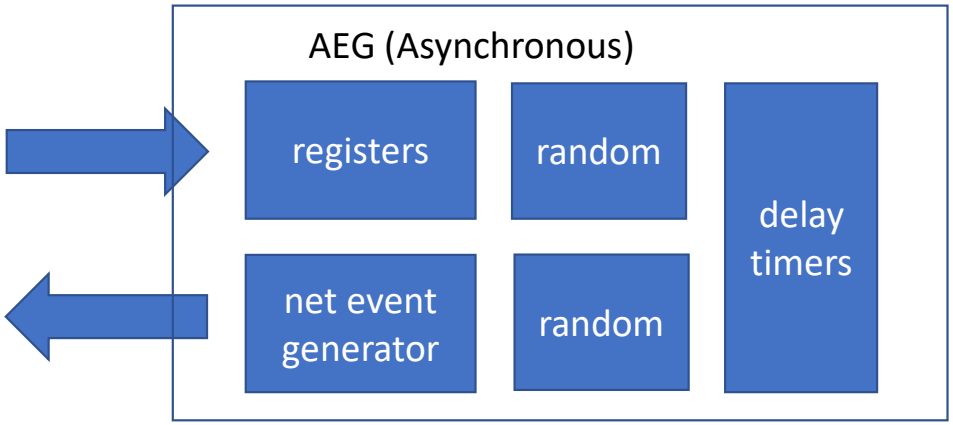
Imperas view:
Components in Async Test Bench (ATB)



- ISG – Instruction Stream Generator
- SEG – Synchronous Event Generator
- AEG – Asynchronous Event Generator
- DUT –the RTL of the RISC-V CPU Device Under Test
- CRM – the Controllable Reference Model (e.g. Imperas model)
- ISP – IntroSPector (tracer) to provide details of what happens in the micro-architecture of the DUT
- SYN - SYNchronizer to keep CRM in sync with DUT
- COM - COMparator between DUT and CRM
- SCB – SCoreBoard to track what has been 'seen' (instructions/events/modes)

=> Complete CPU HW DV environment for best results

AEG – Asynchronous Event Generator



- software writes ranges, and list of nets
- event delay determined by random
  - Delay can be value from test program
- net choice selection determined by random
  - Choice can be from test program

- Connected to DUT (interrupt) pins in testbench
  - Maybe via CLIC…
  - Maybe debug…
- Programmed by test program running on DUT
  - Write to register time delay, net to affect
  - After delay net event takes place
  - Can have simultaneous events

For testing…

- Testbench has SystemVerilog virtual peripheral module instanced
- Imperas ISS has C virtual peripheral instanced
- Same programmers view and functionality… (designed to same spec)
  - And use macros in test programs so different designs can be different
- Program configures and drives async event gen (virtual peripheral)
- And then user chooses how much comparison to do…
  - Go / No-go – did the event occur (e.g. simple compliance testing)
  - Detailed – micro-architecture verification…
- QED

Notes on Interrupt functionality testing

Simon Davidmann, simond@imperas.com

24 June 2021