

XIF Verif Cooperation

Robin Pedersen <u>robin.pedersen@silabs.com</u>



Context



- Shared interest in XIF
 - (CSEM, Silabs, Thales, Dolphin?, more?)
- Potential for cooperation
- How to orchestrate?



Outline



- Architectures & Proposals
- Challenges
- (Discussions throughout)

© OpenHW Group



XIF Primer (quick refresher)



Compressed interface		
Issue interface		
Commit interface		
Memory (request/response) interface		
Memory result interface		
Result interface		

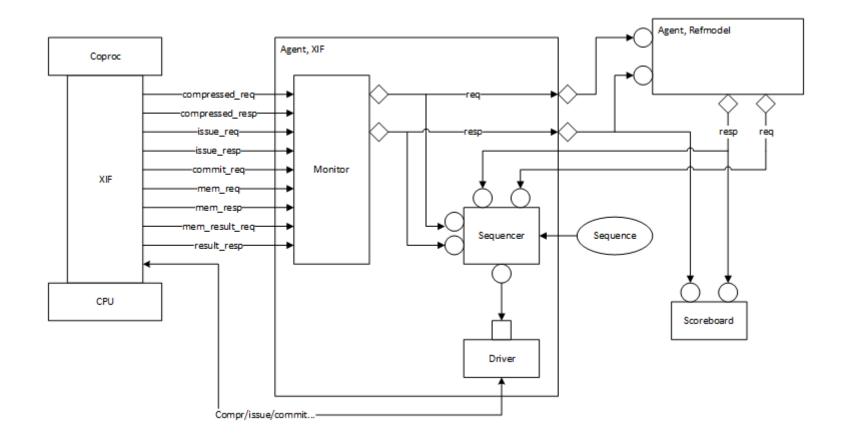
Signal	Туре	Direction (CPU)
issue_valid	logic	output
issue_ready	logic	input
issue_req	x_issue_req_t	output
issue_resp	x_issue_resp_t	input

Signal	Туре	Description
instr	logic [31:0]	Offloaded instruction.
mode	logic [1:0]	Privilege level (2'b00 = User, 2'b01 = Supervisor, 2'b10
id	logic [X_ID_WIDTH-1:0]	Identification of the offloaded instruction.
rs[X_NUM_RS-1:0]	logic [X_RFR_WIDTH-1:0]	Register file source operands for the offloaded instruc
rs_valid	logic [X_NUM_RS-1:0]	Validity of the register file source operand(s).
ecs	logic [5:0]	Extension Context Status ({ mstatus.xs ,``mstatus.fs``,
ecs_valid	logic	Validity of the Extension Context Status.
<		>



Draft 1/4 – "Full" Agent, use-case-configurable

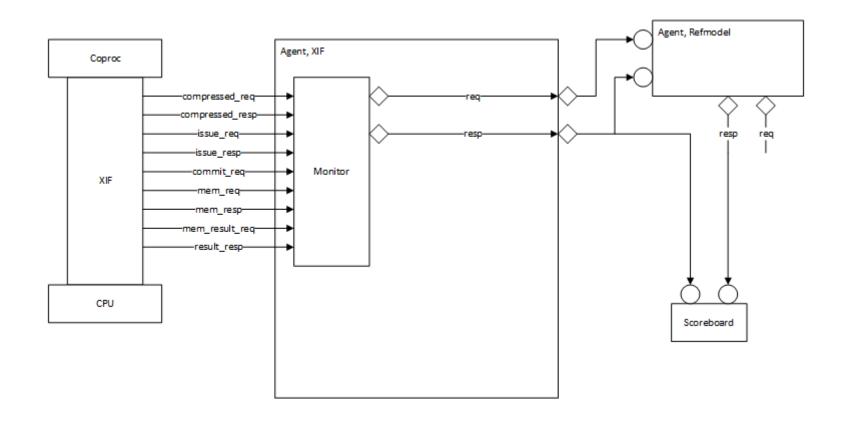






Draft 2/4 – Passive Agent

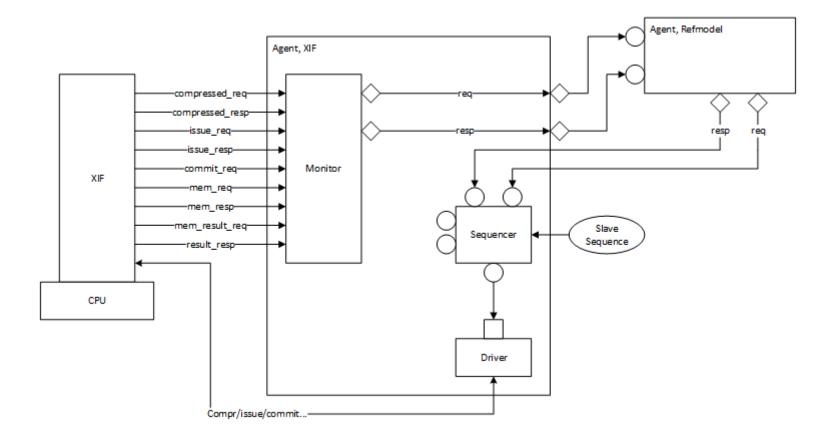






Draft 3/4 – Active Slave Agent (exercise cpu)

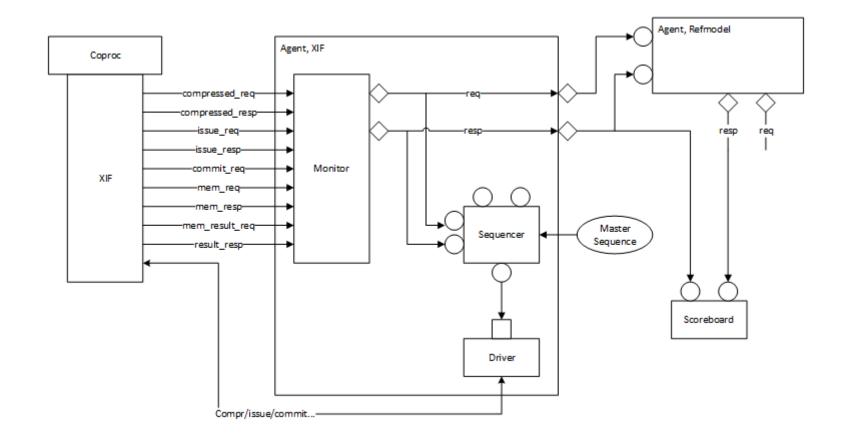






Draft 4/4 – Active Master Agent (exercise coproc)

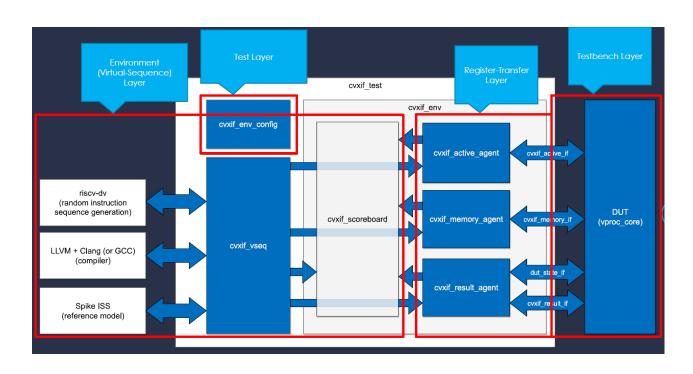


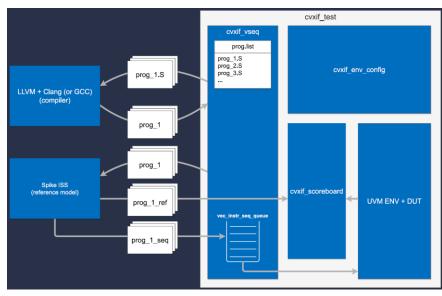




CSEM 1/3 - Environment



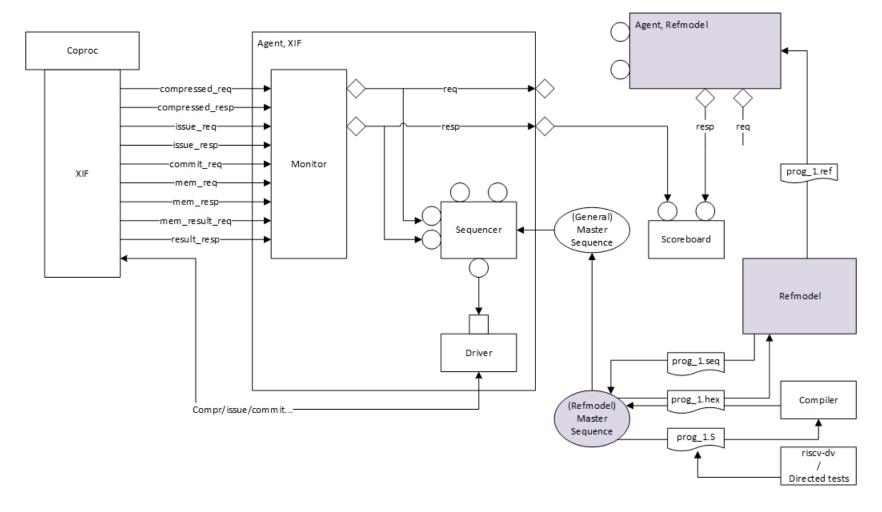






CSEM 2/3 – Potential Integration A

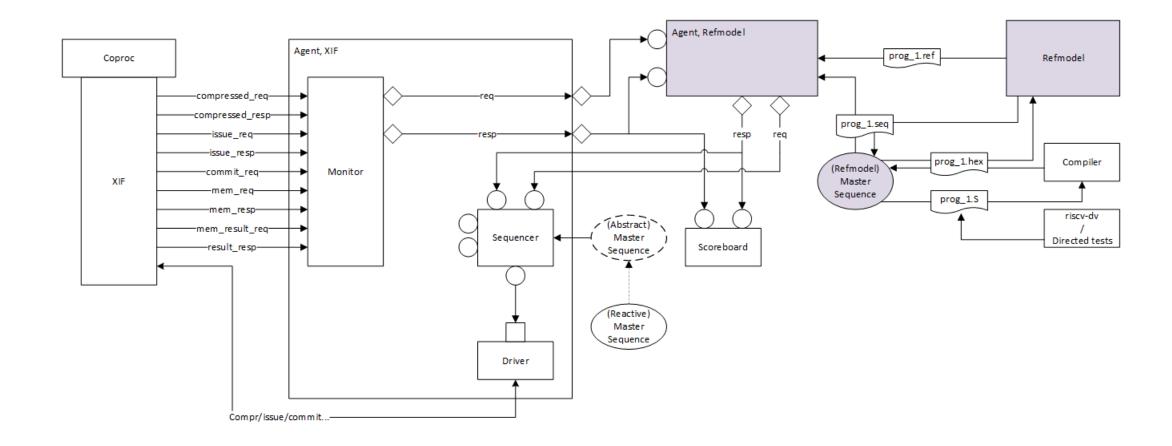






CSEM 3/3 – Potential Integration B

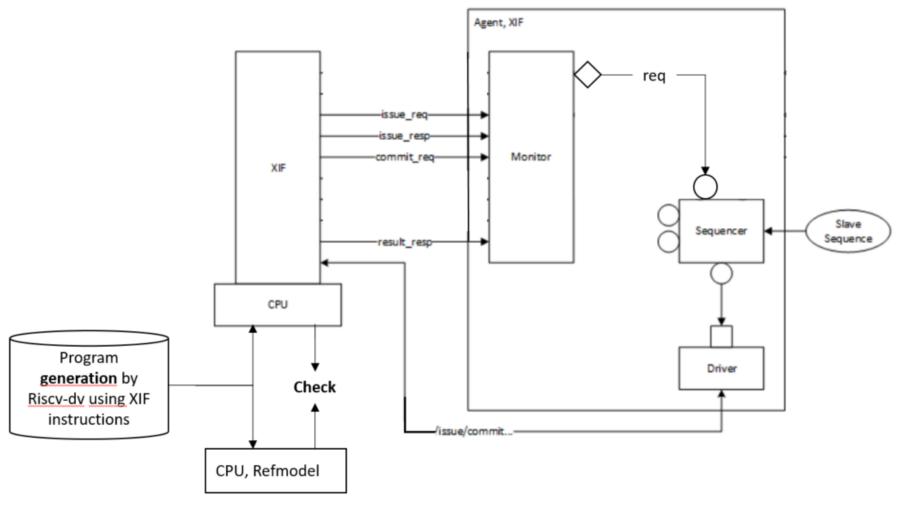






Thales 1/2 - Environment

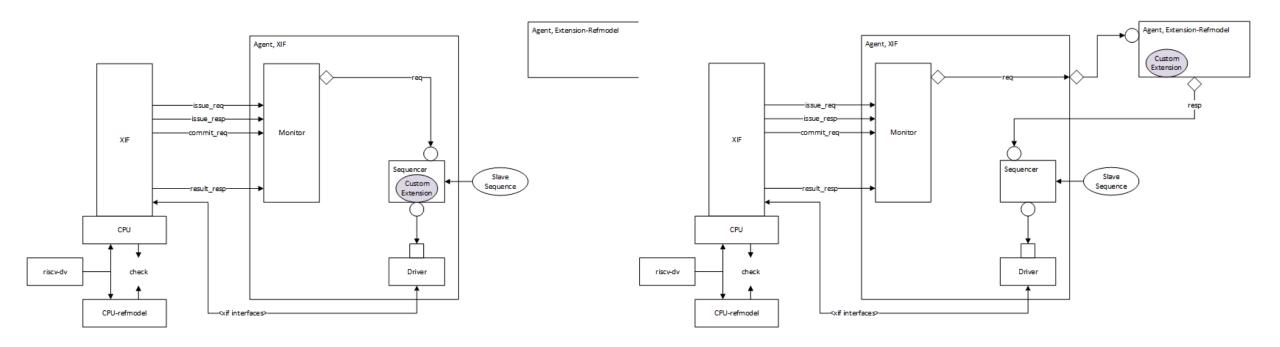






Thales 2/2 – Potential Migration Route







Challenges 1/8



14

- The next slides:
 - Challenging aspects, needs decision making
 - (They all relate, so the order isn't too significant)



Challenge 2/8— Sub-interface Agents?



15

- One agent per XIF "sub-interface"?
 - E.g. {"active", "memory", "result"}
- Pros:
 - ?
 - More files, cleaner separation
- Cons:
 - How to cross-communicate info?
 - Virtual sequence coordinates?
 - No need to communicate between these agents?
 - More files, more clutter?

Compressed interface

Issue interface

Commit interface

Memory (request/response) interface

Memory result interface

Result interface



Challenge 3/8 – Env/Agent



16

- Ideal:
 - All parties use same env and agent
- Compromise:
 - Multiple envs, shared agent
- Worst case:
 - Total isolation, multiple envs, multiple agents, no coop



Challenge 4/8 – Protocol/Isa



17

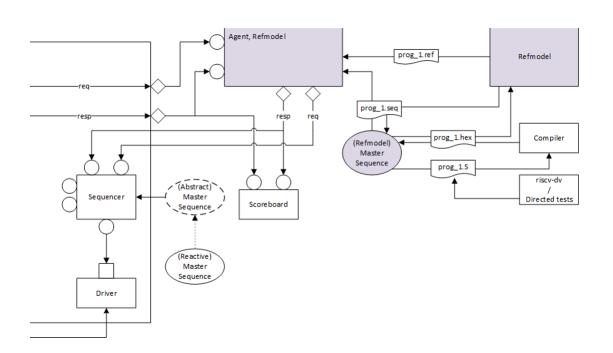
- XIF protocol ≠ ISA extension
 - These are separate concerns
 - Code could reflect separation, or bundle it
- Ideal:
 - Agent is isa-extension-agnostic
 - "add-on-based"
- Worst case:
 - Every ISA-extension has its own XIF agent, no shared code

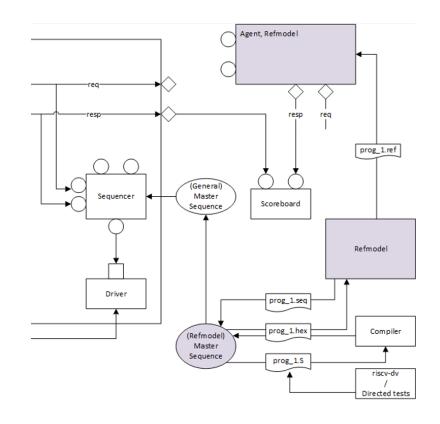


Challenge 5/8 – Stimuli/Checking



- Clear separation of stimuli/checking?
 - Or clear separation of protocol/isa?
 - Mutually exclusive?



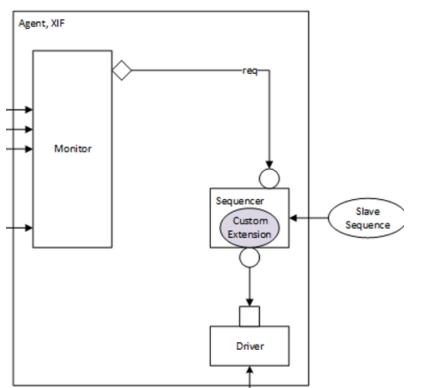


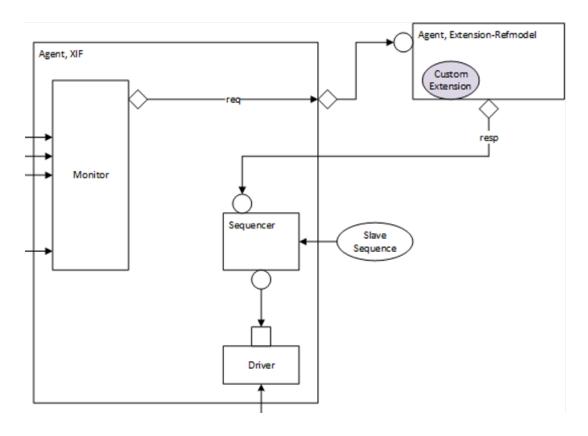


Challenge 6/8 – Inheritance or two agents?



- Connect new "refmodel agent" to add new ISA-extension?
- Derive new "xif agent" to add new ISA-extension?









Challenge 7/8 – Vseq/TLM



20

- Assuming two agents (xif agt, refmodel agt)
 - Use "virtual sequence" for refmodel-agt to xif-agt control?
 - Use "TLM ports" for refmodel-agt to xif-agt control?



Challenge 8/8 – Core-v-verif?



- Pros:
 - All parties can easier collaborate and use the agent/env
 - OpenHW doesn't need a new repo (for csem files)
 - ?
- Cons:
 - Git history from previous repo (lost upon migrating?)
 - Porting effort?
- Misc:
 - Good opportunity for trying out repo re-org?
 - "core-v-agents", or "core-v-lib"?



Discussion



•



Other Topics

SORE-V

- SVAs
- Non-XIF topics
- Ad-hoc topics?





