SoC Firmware Debugging Tracer in Emulation Platform

Kubendra Kumbar [1]
SSIR- MST
Samsung Semiconductor India
Bangalore India
kubendra.sdp@samsung.com

Sandeep Vallabhaneni 2
Controller Development Team
Samsung Semiconductor
Suwon South Korea
v.sandeep@samsung.com

Ken Joseph Kannampuzha[3]
SSIR –MST
Samsung Semiconductor India
Bangalore India
ken.joseph@samsung.com

Hojun Shim [4]
Controller Development Team
Samsung Semiconductor
Suwon South Korea
hojun.shim@samsung.com

Byung Chul Yoo [5]
Controller Development Team
Samsung Semiconductor
Suwon South Korea
byung.yoo@samsung.com

Why SoC Firmware Debugging Tracer

To address the Firmware debugging challenges and performance analysis in the System on Chip (SoC) designs .
Current co-verification solutions are

- ❑ Not compatible
- ❑ Infeasible
- ❑ Expensive
- ❑ Takes more gates counts with SoC platform prototypes , unavailability of required system design debugging due to limitation of breakpoint feature like with In-ciruit Emulation(ICE)
- ❑ Software debuggers and also not capture the exact dynamics of the system due to its limitation ,hence it is difficult to reproduce the exact bugs or fixing the issue .

An optimal and efficient SoC firmware tracer mechanism using firmware tracer fine-tuned to PXP emulation platform.

SoC based Storage Controller

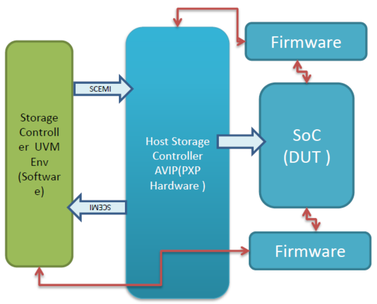

Figure : SoC based Storage Controller

In our current flash SoC controller design ,

1. We are not able to add the Advance RISC Machines (ARM) based embedded Trace module (ETM) for the cortex based M3& M7 processors debugging in chip design .

2. also in PXP emulation platform due to chip area , cost considerations and resource crunch in the emulators .

To overcome the above problem and alternative debugging trace solution is proposed with less gate count and cost consideration .

Firmware Debug Tracer

- ❑ The same way of waveforms debugging in the hardware debugging is adopted for firmware debugging for the easy way to debugging .
- ❑ This feasible solution is developed for the Cortex M3/M7 series processors and it helps for he where it get struck and also where we can optimize the firmware functions by plotting in waveforms of the function name.

➢ ie Done with cadence simvision (RTL simulation tool) .

- ❑ This firmware debug tracer will debug the software/firmware component that enables the reconstruction of program execution of each firmware functions on the simulation waveform.

- ❑ Helps in optimizing the firmware components & also used as feedback for the next chip tape outs /revisions for the better performance by changing in the architectural

Implementation

This is implemented by current firmware function that is being executed and the hierarchical level of the Firmware functions under execution plot with help of corresponding Program Counter (PC) value and searching this function name based on the function address

To implement the Firmware Tracer,

- ❑ we need to know the information about the function names for every corresponding Program Counter value with respective cortex M3/M7 processors .

- ❑ also to predict the correct program counter value depending on the PUSH/POP conditions considering the

✓ Branch and Branch booster conditions,
✓ ISR interrupts
✓ Branch with link instruction etc.

- ❑ This function name and function address is extracted from the given firmware code using Perl scripts and system level addressing.

- ❑ For this searching logic, we can use a for-loop to search the function name based on the function address and but requires a huge number of resources in Hardware (approximately 800K logic gates for 900 functions) , to save logic gates, hardware & software implement ion using SCEMI-pipe /DPI/ MARG(Cadence specific interface) methods are used .

Implementation –Continued

➢Firmware Tracer architecture using Standard Co-Emulation Modelling Interface (SCEMI) pipes show below .

➢This FW tracer makes use of SCEMI-pipes to send the program counter value to the Software part and get back the function name based on program counter value.

➢This communication must be allowed to happen only during waveform dumping; otherwise, this SW to HW communication becomes an overhead and degrades the simulation speed performance .

➢A generic data file (DAT) is created for the given unit level/product level firmware with respect to function names & function address using the script language. After the loading the real firmware code in terms of ITCM (Instruction tightly coupled memory), DTCM (Data Tightly coupled-Memory) and VROM into the test bench.

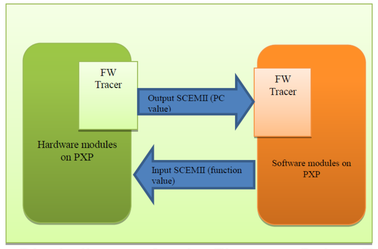

Figure : Block diagram for the Firmware Debug Tracer

Basic operation & Waveforms

Each functions of firmware will be displayed in the waveform with help of PC value of the processors

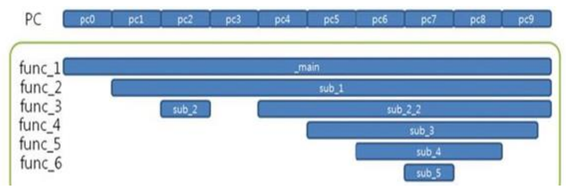

Figure : Firmware Tracer Operation

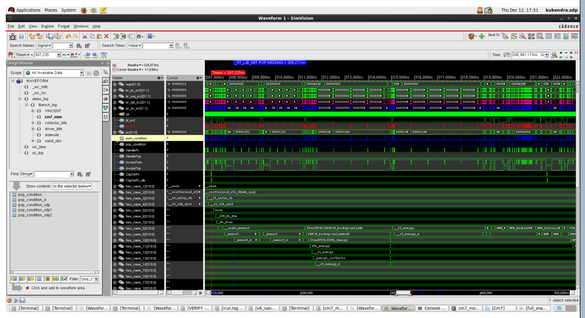

Figure : Simulation results for the PUSH & POP Conditions with function name & PC value

Implemention-Contineed

- ❑ Here we use the SCEMI-pipes to send the program counter value to the Software(S/W) part and get back the function name based on program counter value

- ❑ Define SCEMI output pipe/MARG interface on HDL-side to send the Function_Label_Addr[31:0] and PC value from HW side to SW side.

//Using the blocking send method, PC value to software module for the function name searching.

- ❑ Define SCEMI input pipe on HDL-side to receive the Function_Label_Name[319:0] from SW side to HW side. Using the blocking receive method, function name is received from the S/W module for the plotting waveform
//The size of the function name is changed depending upon the requirement of projects A switch is used to have configurable size for the function name.
- ❑ Create an System Verilog (SV) file with sv input and output scemi pipes and connect these pipes to the SCEMI pipes on the HDL side.
- ❑ Add logic in SV file to get the function address from HDL side using SCEMI output pipe, find the corresponding function name and send it to the HDL side using SCEMI input pipe.
// Hardware Part: Glue logic for the finding the PC value depending upon various PUSH/POP conditions using
// cortex M7 signals like bl_ext2 , br_ex2 , int_exit , int_entry ,pc_ret,_rege ,pop_to_pc_iss and br_pc_ex2 etc .
// Software Part: searching logic (using the for loop) for the function names depending upon the provided PC value

- ❑ Call the send and receive tasks of SCEMI pipes on HDL side FW tracer during waveform dump.

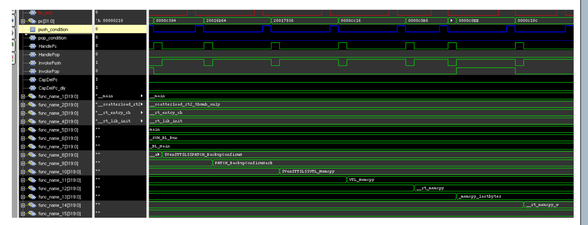

Figure : Simulation results for the PUSH Conditions with function name & PC value

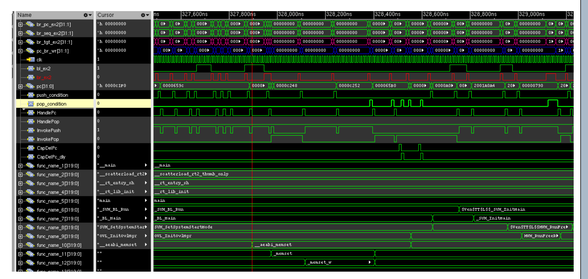

Figure : Simulation results for the POP Conditions with function name & PC value

Conclusion

▪This is cost effective in terms of cost and gate count ;firmware tracer ;can easily debug the firmware code, optimize the software and hardware blocks and can switch logic (from hardware and software and vice versa) for the better performance results.
▪For the Best speed & performance is achieved speed by doing "function name searching logic with software part and Program counter value logic with hardware part using the SCEMI pipe based interface.

Current Challenges

•The PC value predicted will vary from processor to processor.
•Depending upon the tradeoff between gate count and speed, emulation hardware and software optimization is done by switching the Hardware/Software logic using the DPI/SCEMI /MARG mechanisms in the Emulation