

A case study into improving Network device driver quality by design



Anjali Singhai Jain

Kevin C Scott

www.intel.com

PNSQC 2012

Goal: Have software written and validated before silicon arrival.

Question: How?

Motivation

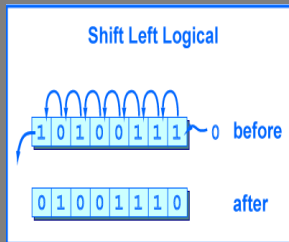
- Reduce the number of “silly” bugs and code duplication
- Allow greater amount of time to focus on:
 - Test scenarios
 - Performance
 - Ecosystem enablement
 - Time to pay attention to use cases

40 Gigabit Ethernet Driver

- Windows, Linux, Solaris, FreeBSD
- Virtualization: ESX, Zen, KVM, Windows Hyper-V
- Storage Drivers (FCoE)
- iWARP Drivers
- Firmware and Tools Libraries



Design Philosophy



Shift Left



Sharing & Collaboration



Automatic code generation

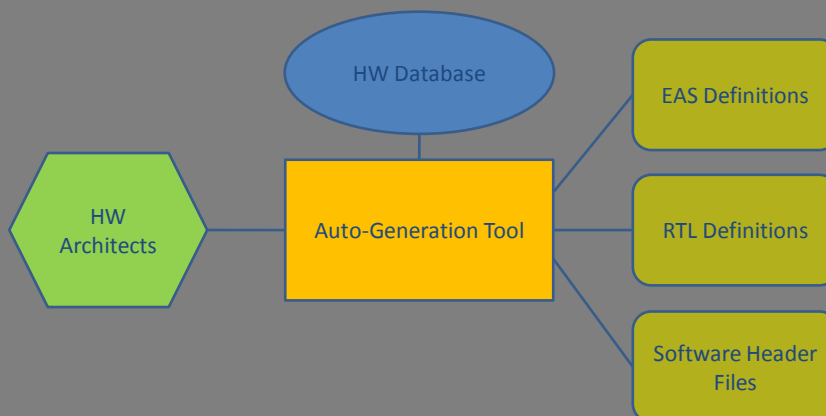
Hardware Emulation

Early validation with hardware and firmware teams

Generic Client Interface for protocol drivers

5

Automatic Code Generation



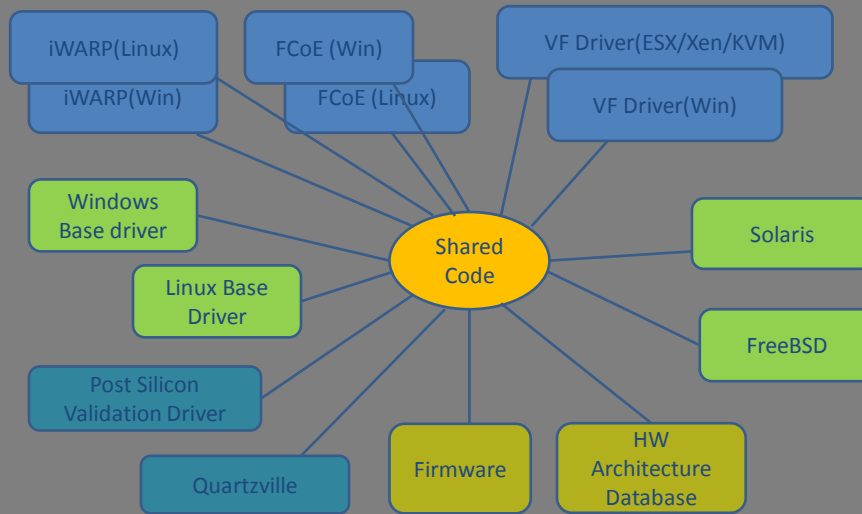
Benefits and Future work

- Avoid human errors in EAS interpretation
- Maintain consistent coding style
- Save ton of time for a product as huge as 40 Gig NIC
- Always up to date with what is in the EAS
- Side Effects:
 - More free time to Socialize with your co-worker 😊

Firmware Team Collaboration

- 40Gig SW drivers have a much larger dependency on FW
- Due to SW mandate to Shift Left, Firmware needs to do the same
- A very early collaboration was needed
- Co-developed the Interface
- Firmware Header file:
 - Follows the shared code style guideline
 - Patches generated by Firmware team
 - Reviewed by SW team
 - Maintained by SW Shared code team
 - Consumers are both SW and FW teams

Fortville Software Components



9

OS abstraction in the shared code

- Abstraction done for
 - Virtually and physically contiguous DMA-able memory allocation.
 - Memory manipulation functions
 - Code Synchronization: Spinlocks
 - Delays
- Hot path vs. Non-Hot path code
- Made possible by collaboration between NDIS and Linux teams

Future Considerations

- Non performance OS abstracted driver completely written in shared code
- Functional Debug driver
- Ground work for “VGA” driver

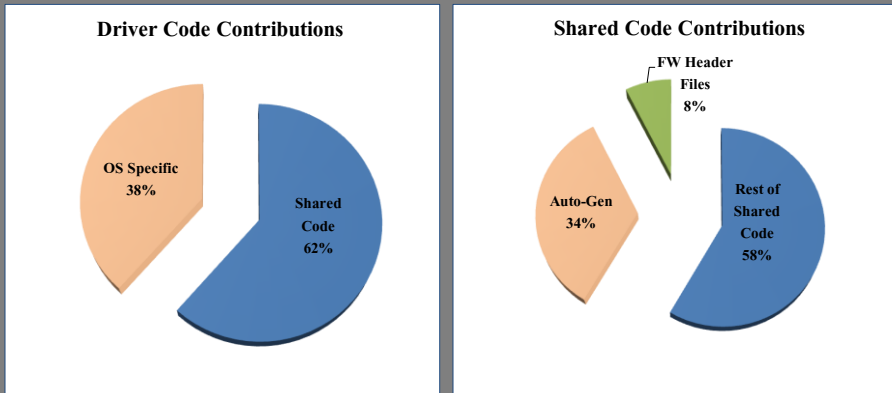
11

Shared Code used by Post Silicon Validation Driver

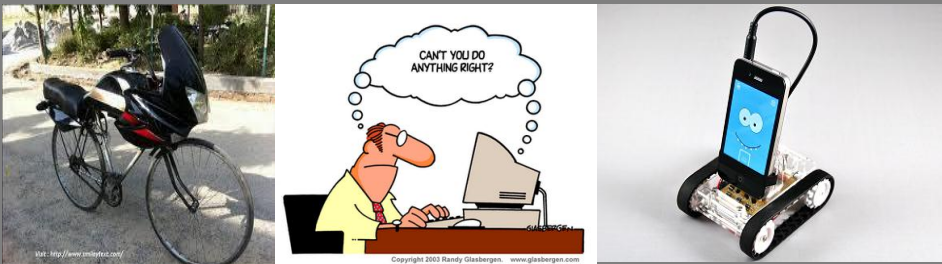
- Shared code is tested before SW driver bring-up on FPGA
- SV Tests are comprehensive, automatic and randomly generated and exercise HW boundaries
 - Provide better coverage for shared code than what unit tests and SW validation can do

12

Where did we reach...



SW Emulation



- **KVM-QEMU Based**
- **Functional Emulation**

Conclusion

- Complexity and the Shift Left challenge helped us rethink and innovate the way we do software
 - New Partnerships
 - New ways to collaborate
 - Redesign SW components to be modular and flexible

15

Questions?