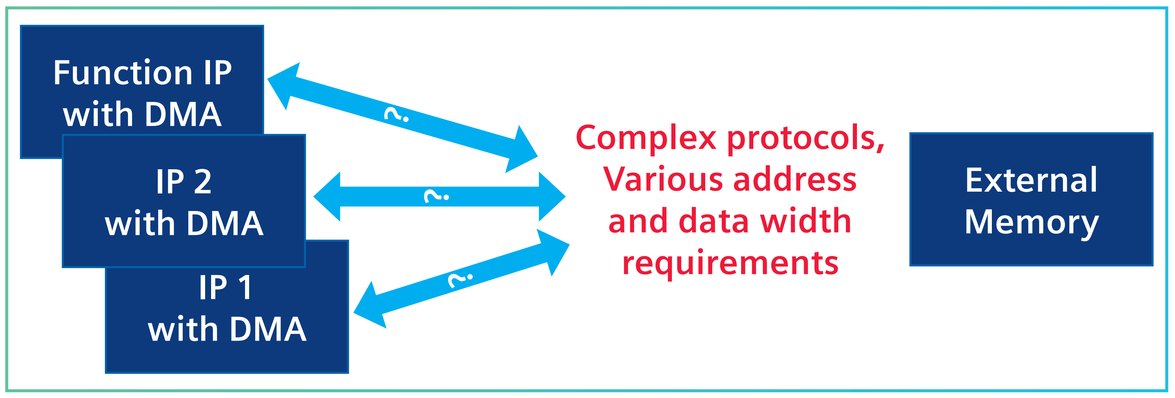
Design and Verification of AXI4 Master for ASIC Using High-Level Synthesis in C++

Telechips { Dooyoung Go and Moonsoo Kim }
Siemens EDA { William Lee }

Introduction

- Direct memory access (**DMA**) can be required for hardware blocks that access external **memory**.
- However, it may be time-consuming to design and verify a DMA block which meets complex bus protocol specifications such as **AMBA AXI** for various address and data bus widths required by different hardware IPs.
- A BUS master IP can be useful for providing **software-like memory access** while achieving **high throughput** using functionalities like burst.
- **C++ HLS** can be adopted for configurability and reusability.



Main Idea – C++ based HLS Design & Verification Flow

```
#pragma hls_design interface
void CCS_BLOCK(run)( ac_channel<AXI4_AW_CHANNEL<AW, IW, UW>>  &ch_AW,\
                     ac_channel<AXI4_W_CHANNEL<BW, IW, UW>>   &ch_W, \
                     ac_channel<AXI4_B_CHANNEL<IW, UW>>       &ch_B, \
                     nel<AXI4_REQUEST<AW, IW, UW>>            &wreq, \
                     nel<AXI4_DATA_W_FLAG<DW, IW, UW>>        &wdata,\
                     nel<AXI4_RESPONSE<IW, UW>>               &wresp,\
```

```
template<int AW, int BW, int IW = 1, int UW = 1>
class tc_axi_master_write {
    axi4_write_requester<AW, BW, DW, IW, UW> requester;
    axi4_writer<BW, DW, IW, UW> writer;
    axi4_responsor<IW, UW> responsor;
    ac_channel<AXI_PACKET_INFO<BW, IW>> if_winfo;
```

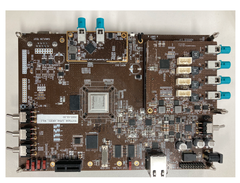
- ac_channel datatype is used for ready-valid handshake channels
  - Algorithmic C **ac_channel** is an opensource HLS library for **handshake interfaces**.
  - **AXI4 channels** (AW, W, B, R, AR) can be easily implemented by using ac_channel.
  - No need to worry about bugs related to handshaking.
- Telechips AXI4 C++ Class
  - Parameters such as address and data bus widths can be **templatized**.
  - Variables related to bit counts can be declared using **template** arguments and **log** functions.
  - High level synthesis can give **different scheduling** results (RTL) for different parameters.
  - **C++ verification** is performed for the same code with a variety of template arguments.
  - Meet AXI4 protocol specs (8-1024-bit bus, **MOR** (<=64), and **burst** (incremental only))
  - DMA input bit <= data bus width (narrow burst).
  - Can automatically split bursts (i.e., if max_burst = 4, a 32 burst is split into 8x 4 bursts).

Main Idea – Templatized C++ Class Using ac_channel

- AXI4 master is designed in **synthesizable C++** code.
  - **ac_channel** and **templates** are used. (Telechips AXI4 class)
- C++ simulation is performed to verify if the design meets the given specification.
  - Most bugs are fixed in this phase using very fast, untimed C++ simulation (**~100x faster** than RTL simulation).
- RTL is synthesized from the **C++ code**.
  - RTL is designed by high level synthesis (Siemens **Catapult HLS**).
  - Architecture exploration for performance (latency, throughput) and area under timing constraints.
- **RTL** verification is performed.
  - All test conditions should be the same as the C++ simulations.
- Check if the results of **C++ and RTL** verifications are identical.
  - Focus on bugs that can only occur in cycle accurate RTL simulations, such as **deadlocks**.

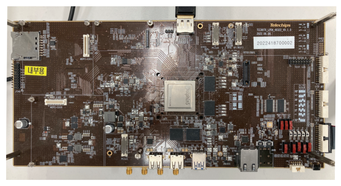
Results

N-Dolphin
NPU based ADAS
(Samsung 14nm)



N-Dolphin
evaluation board

Dolphin 5
Automotive AP
(Samsung 8nm)



Dolphin 5
evaluation board

- A novice hardware engineer designed and verified the AXI4 master in ~1.5 months.
- It can provide software-like memory access to DMAs.
- Multiple outstanding and burst are also supported for high data throughput.
- AXI4 master has been applied to audio processing IP in our automotive MCU (**VCP**, Samsung 28 nm), on-the-fly dewarping engine in AI accelerator chip (**N-Dolphin**, NPU-based ADAS application, Samsung 14 nm), and surround view engine in automotive application processor (**Dolphin 5**, Samsung 8nm), which has been verified through post-silicon verification.

Summary

- Our AXI4 Master IP can achieve **high data throughput by multiple outstanding and burst** as well as provide **simple, software-like** access to DMAs.
- In addition, RTLs for **any address and data bus widths can be extracted** from the same code with different **template arguments**.
- Finally, AXI4 master is currently used for a variety of chips and IPs.
  - Automotive MCU (**VCP**, Samsung 28nm)
  - NPU-based ADAS processor (**N-Dolphin**, Samsung 14nm)
  - Automotive application processor (**Dolphin 5**, Samsung 8nm)