**NATIONAL CHENG KUNG UNIVERSITY**

**College of Electrical Engineering and Computer Science**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**VLSI System Design (Graduate Level)**

**Fall 2022**

**Summary of Final Project**

**Please don’t just write yes/no if there need more details,** **and use single-sided printing**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Simulate at SoC(yes/no)** | | | **Yes** | | | | | |
| **Upload 5-min media(yes/no)** | | | **Yes** | | | | | |
| **Basic** | | | | | | | | |
| **MCU** | **Pipeline** | | **Stage** | | **Max working Freq.** | | | **Data Width** |
| **5(5 stage)** | | **80(>80 MHz)** | | | **32(>32 bits)** |
| **Number of Instructions** | | **49(>49)** | | | | | |
| **Realized Cache Specification** | | **L1cache**  **Associative**  **Write through**  **(e.g. L1 size, associative, read/write policy)** | | | | | |
| **Cache Hit Rate of each program** | | **Rtl0: l1i:0.992 l1d:0**  **Rtl1: l1i:0.999 l1d:0**  **Conv0: l1i:0.999 l1d:0**  **Conv1: l1i:0.999 l1d:0**  **Conv2: l1i:0.999 l1d:0**  **Pool0: l1i:0.999 l1d:0**  **Convall: l1i:0.999 l1d:0** | | | | | |
| **List of Realized Forwarding in Types and Stages** | | **MEM to EXE**  **WB to EXE**  **(e.g. which kind)** | | | | | |
| **Realized Performance Counters (IPC) of each program** | | **Rtl0: 0.702**  **Rtl1: 0.799**  **Conv0:0.723**  **Conv1: 0.744**  **Conv2: 0.746**  **Pool0:0.774**  **Convall: 0.753** | | | | | |
| **Interrupt mechanism** | | **YES** | | | | | |
| **Memory** | **On-chip memory**  **(Total size <= 320KB)** | | **IM** | | | **DM** | | |
| **65KB** | | | **65KB** | | |
| **Off-chip memory** | | **SDRAM** | | | **ROM** | | |
| **8MB** | | | **16KB** | | |
| **ASPU** | **Max working Freq.** | | **12.5ns** | | | | | |
| **Processing speed (throughput or… )** | | **FPS=14.624 frame per second** | | | | | |
| **Realized Specification of Functionalities in details** | | **3\*3convolution**  **1\*1convolution**  **Max pooling** | | | | | |
| **Comparison with other works if any** | |  | | | | | |
| **BUS** | **Specify Memory and I/O mapping** | | **Slave** | **Start address** | | | **End address** | |
| **ROM** | **0x00000000** | | | **0x00004000** | |
| **IM** | **0x00010000** | | | **0x0001FFFF** | |
| **DM** | **0x00020000** | | | **0x0002FFFF** | |
| **DRAM** | **0x20000000** | | | **0x207FFFFF** | |
| **SCTRL** | **0x10000000** | | | **0x100003FF** | |
| **WDT** | **0x10010000** | | | **0x100103FF** | |
| **EPU** | **0x50000000** | | | **0x800FFFFF** | |
|  |  | | |  | |
| **Implemented Features of AXI Bus, Level of Realization, Operating Frequency,**  **Outstanding number** | | **80MHz** | | | | | |
| **System** | **Specify** **Cooperation between CPU, Bus, Memory, ASPU and others** | | **Assume ALL input/weight/bias data in DRAM.**  **CPU runs booting program with DMA.**  **Use DMA to move data from DRAM to EPU’s buffer.**  **CPU writes to EPU ctrl registers.**  **EPU writes to output buffer as CPU stuck at WFI(Wait for interrupt).**  **EPU finishes and send interrupt. CPU continues with ISR(Interrupt service routine).**  **CPU writes ctrl signals for next layer.**  **Trigger “In-Output buffer swap”**  **Output of this layer is the input of next layer**  **If done, DMA move data from EPU to DRAM.**  **(e.g. How your system works)** | | | | | |
| **Specify Hardware interrupt & Interrupt service routines** | | **WTO**  **EPU**  **Sensor conctrl**  **(>2 kind, and how they work)** | | | | | |
| **Specify Mechanism for Booting from an external ROM** | | **Move data from DRAM to destination memory** | | | | | |
| **Specify Realized DMA(Direct Memory Access) and Usage** | | **CPU will give DMA source address, length, destination address. After receive this information, DMA moves data from source address to destination address. If done, sending finish to CPU.** | | | | | |
| **Code analysis (Superlint)** | | | **99%(should >99% error free)** | | | | | |
| **System w/ ASPU (yes/no)** | | | **YES** | | | | | |
|  | | | **Synthesis** | | | **APR** | | |
| **clock period** | | | **12.5ns** | | | **12.5ns** | | |
| **Power** | | | **134.6699mW** | | | **304.26555650mW** | | |
| **Area** | | | **47608660** | | | **69447383.21 um^2** | | |
| **Verification** | **MCU** | **prog0 pass ratio** | **100%** | | | | | |
| **ASPU** | **# and types of Direct test or constrained random test** | **Direct test** | | | | | |
| **Specify types, length, operation conditions of benchmarks** | **TB verify EPU each layer output with software result after DMA move output data to DRAM.** | | | | | |
| **S**  **Y**  **S**  **T**  **EM** | **prog0 PR**  **simulation time** | **1614787500 PS** | | | | | |
| **prog1 PR**  **simulation time** | **23219787500 PS** | | | | | |
| **Specify types, length, operation conditions of benchmarks** | **TB verify EPU output with software result after DMA move output data to DRAM.** | | | | | |

|  |  |
| --- | --- |
| **Advanced** | |
| **Synthesize AXI bus with burst and fully work with IPs** |  |
| **30 more instructions** |  |
| **64-bit add/sub, store/load** |  |
| **I/O PADs** |  |
| **More cache (L2 or L3)** |  |
| **dynamic branch prediction** |  |
| **CRT for more than two IPs** |  |
| **floating-point co-processor** |  |
| **Bootable by an operating system** |  |
| **Verify with FPGAs, specify FPGA board, what module has been put on the board and how you confirm results** |  |
| **Other Properties, please specify** |  |
| **References** |  |