FTDL: An FPGA-tailored Architecture for Deep Learning Systems

Runbin Shi, Yuhao Ding, Xuechao Wei, Hang Liu, Hayden So and Caiwen Ding
The University of Hong Kong, Peking University, Stevens Institute of Technology, University of Connecticut
{rshi, yhding, hso}@eee.hku.hk, xuechao@pku.edu.cn, hliu77@stevens.edu, caiwen.ding@uconn.edu

Motivation

Among the existing FPGA deep learning (DL) accelerators, most of them deployed application-specific integrated circuit (ASIC)-oriented architectures to FPGA without considering the FPGA underlying layout, which leads to the architecture-layout mismatch. Such a mismatch results to a low $f_{\text{max}}$ in implementation ($\approx 200$MHz), while the computational unit (DSP) in FPGA can achieve an $f_{\text{max}} \approx 750$MHz. The contributions of this work are summarized as,

- **Good timing and scalability**: FTDL proposes a novel overlay architecture for convolutional and fully-connected layers that is tailored for the tiled structure of modern FPGAs, allowing post-place-and-route operating frequencies to reach over 88% of the theoretical DSP operating frequency across different devices and design scales.
- **High hardware-efficiency**: FTDL provides a compilation tool that maps most DL-layers to the overlay with over 80% hardware-efficiency on average.

Workload Scheduling

Objectives:

- **Optimal performance**
- **Balance between performance and WBUF efficiency**
- **Optimal hardware configuration**

Problem formulation:

Figure 5. The tiled loops represent the workload scheduling in spatial and temporal; The trips counts of sub-loops compose the mapping vector. Note that both CONV and MM are analyzed as a K-level nested loop.

Top-200 optimal solutions:

Figure 6. Roofline-based visualization tool for performance analysis. (a) and (b) plot top-200 optimal solutions by FTDL compiler for performance and balance objectives respectively. The solution in (b) is preferable as they saves WBUF $5 \times$ to (a) with only slight performance loss. Note that the y-axis has been scaled to the area of interest.

FTDL: An FPGA-tailored Architecture for Deep Learning Systems

Runbin Shi, Yuhao Ding, Xuechao Wei, Hang Liu, Hayden So and Caiwen Ding
The University of Hong Kong, Peking University, Stevens Institute of Technology, University of Connecticut
{rshi, yhding, hso}@eee.hku.hk, xuechao@pku.edu.cn, hliu77@stevens.edu, caiwen.ding@uconn.edu

Motivation

Among the existing FPGA deep learning (DL) accelerators, most of them deployed application-specific integrated circuit (ASIC)-oriented architectures to FPGA without considering the FPGA underlying layout, which leads to the architecture-layout mismatch. Such a mismatch results to a low $f_{\text{max}}$ in implementation ($\approx 200$MHz), while the computational unit (DSP) in FPGA can achieve an $f_{\text{max}} \approx 750$MHz. The contributions of this work are summarized as,

- **Good timing and scalability**: FTDL proposes a novel overlay architecture for convolutional and fully-connected layers that is tailored for the tiled structure of modern FPGAs, allowing post-place-and-route operating frequencies to reach over 88% of the theoretical DSP operating frequency across different devices and design scales.
- **High hardware-efficiency**: FTDL provides a compilation tool that maps most DL-layers to the overlay with over 80% hardware-efficiency on average.

Workload Scheduling

Objectives:

- **Optimal performance**
- **Balance between performance and WBUF efficiency**
- **Optimal hardware configuration**

Problem formulation:

Figure 5. The tiled loops represent the workload scheduling in spatial and temporal; The trips counts of sub-loops compose the mapping vector. Note that both CONV and MM are analyzed as a K-level nested loop.

Top-200 optimal solutions:

Figure 6. Roofline-based visualization tool for performance analysis. (a) and (b) plot top-200 optimal solutions by FTDL compiler for performance and balance objectives respectively. The solution in (b) is preferable as they saves WBUF $5 \times$ to (a) with only slight performance loss. Note that the y-axis has been scaled to the area of interest.

References