# Design of an Intelligent DMA System Architecture with Data Pre-processing

Pengyi Wu[1, a], Zhaolin Li[2, b]

[1] Institute of Microelectronics, Tsinghua University Beijing, 100084, China

[2] Research Institute of Information Technology, Tsinghua University Beijing, 100084, China

[a] wpy13@mails.tsinghua.edu.cn, [b] lzl73@mail.tsinghua.edu.cn

**Keywords:** DSP; DMA; PRAM; Floating- point computation; Pre-computation

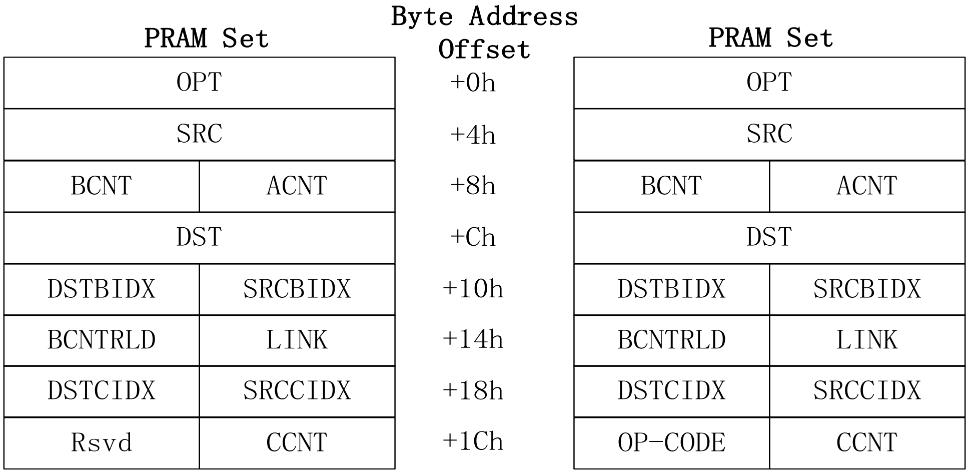
**Abstract.** DMA is often used to transfer data between different kinds of memory-mapped slave endpoints. It is a common peripheral component for DSP. In order to make full use of the time during the reading and writing, this paper proposed an intelligent DMA. Compared with the traditional DMA, a newly designed operational unit is added to the RAM-based architecture and the parameter RAM(PRAM) is improved to suits the whole DMA controller. It supports floating-point computation and the interface is designed based on the AMBA bus protocol AXI3. FFT algorithm has been used to evaluate the proposed design.

## Introduction

With the development of integrated circuits and the advancement in semiconductor progress, a variety of bus protocols have been proposed among which AMBA bus architecture showed advantages among others because of its low power consumption and compatibility. DMA has been proposed with the demand for the amount and the speed of processing data. It saves the workload of CPU and improves the efficiency of the whole system.

DMA must be constantly optimized to guarantee the work of DSPs such as image processing, target tracking, radar signal analysis, scientific computing etc. [1][2] TMS320 C2000, C5000 and C6000 are three products of DSPs produced by Texas Instruments (TI) which are mainly used in area of digital control, communication and portable application, audio and video technology respectively[3][4][5]. There DMAs are optimized to supports there work. However, there DMAs do not have operational units. Efficient could be largely improved if data have been pre-processed before arriving at the CPU core.

## Design of the Intelligent DMA



| PRAM Set | | Byte Address Offset | PRAM Set | |
|---|---|---|---|---|
| OPT | | +0h | OPT | |
| SRC | | +4h | SRC | |
| BCNT | ACNT | +8h | BCNT | ACNT |
| DST | | +Ch | DST | |
| DSTBIDX | SRCBIDX | +10h | DSTBIDX | SRCBIDX |
| BCNTRLD | LINK | +14h | BCNTRLD | LINK |
| DSTCIDX | SRCCIDX | +18h | DSTCIDX | SRCCIDX |
| Rsvd | CCNT | +1Ch | OP-CODE | CCNT |

Fig. 1 Parameter RAM set

(a)Traditional one (b) improved one

**The traditional DMA.** The DMA consists of two blocks: channel controller and transfer controller. The main blocks of the channel controller are DMA/QDMA channel logic, parameter RAM(PRAM), event queues, transfer request submission logic, completion detection and registers[6]. The main blocks of the transfer controller are DMA program register set, DMA source active register set, read controller, destination FIFO register set, write controller, data FIFO and completion interface. This paper designs a new PRAM format in channel controller and an operational unit which supports floating-point computation.

**The PRAM.** The DMA controller is a RAM-based architecture. As the traditional PRAM shows in figure1, the PRAM contains transfer parameters such as source address, destination address, transfer counts, indexes, options etc. [7] Each PRAM set includes eight 4-byte set entries and the first two bytes of the last 4-byte entry is reserved. So, this RAM set can be improved according to the different needs.

Talbe1 and figure2 clearly illustrates the 16-bit OP-CODE. Op shows the operational mode. Times could be the count when DSP wants to execute the operation of accumulation and the address is the place where the operand be placed in the data FIFO.

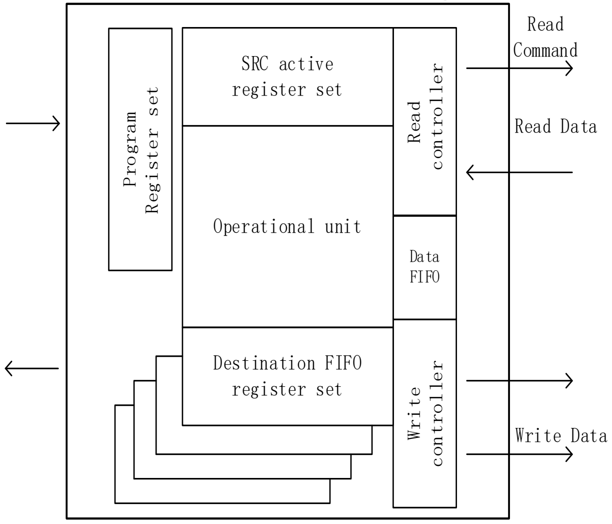
Table 1.OP-CODE description

| Bit | Field | Description |
|---|---|---|
| 15~12 | Op | 15:Floating point or fixed point<br>14:Accumulation or add a number in the specified address<br>13~12:00,operation add;01,operation multiply<br>10,operation accumulation;<br>11,operation power sum |
| 11~8 | Times | The count of numbers in accumulation |
| 7~0 | Addr | The address of the operand |

| 15     12 | 11     8 | 7     0 |
|---|---|---|
| Op | Times | Addr |

Fig. 2 OP-CODE

**The operational unit.** Figure.3 shows the block diagram of the DMA transfer controller. Figure.4 illustrates the operational unit in figure.3.



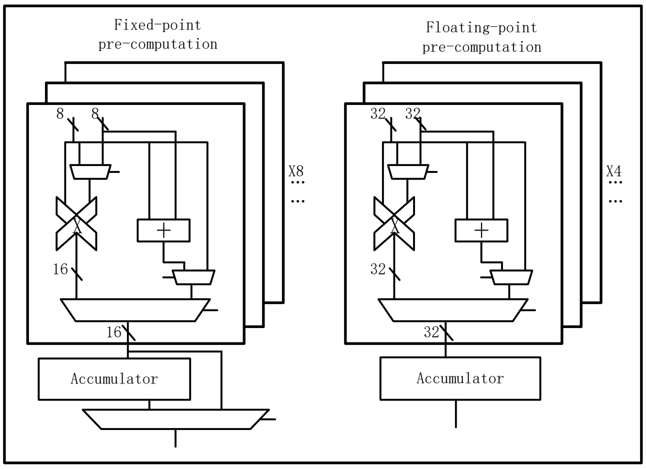


Fig. 3 DMA transfer controller block diagram

Fig. 4 Operational unit

If data FIFO has space available for the data, the read controller issues read commands and the write controller starts issuing write commands as soon as there is sufficient data in the data FIFO. The

DSTREGDEPTH parameter determines the number of entries in the DST FIFO register set which determines the amount of TR pipelining possible for a given TC. This means that read controller can go ahead and issue read commands for the subsequent TRs while the DST FIFO register set manages the write commands and data for the previous TR. The value of DSTREGDEPTH can be set as n, so the read controller can process up to n TRs ahead of the write controller[7].

The operational unit consists of two parts, floating-point pre-computation and fixed-point pre-computation. The part of fixed-point pre-computation is mainly aimed at radar signal processing. Radar signals are represented by grey value so all the adders and the multipliers in fixed-point part are eight-bit[8]. The part of floating-point pre-computation consists of adders and multipliers with 32-bit input. There are sixteen fixed-point adders, sixteen fixed-point multipliers, seven floating-point adders and four floating-point multipliers integrated in the operational unit. The floating-point adders and multipliers are designed using 3 cycles of pipeline. As it is shown in figure 4, the floating-point precomputation part supports accumulation up to 4 operands. So it must go through another two operation of add in accumulator. However, some of the operations except the accumulation and power sum in fixed-point can be completed in one cycle.

## Experiment and result

In order to evaluate the area and power, the proposed DMA design has been modeled by Verilog HDL and synthesized in a TSMC 65nm CMOS technology after fully functional verification.

**FFT Evaluation.** Fast Fourier transform (FFT) is a high-efficient algorithm of discrete Fourier transform. It is improved based on the Fourier transform’s characteristics, parity, symmetry of real part and imaginary part.

As it is clearly shown in figure.5, there must be a four-number-accumulation in every step of the butterfly operation. The proposed DMA could complete this accumulation if the PRAM is set appropriately.

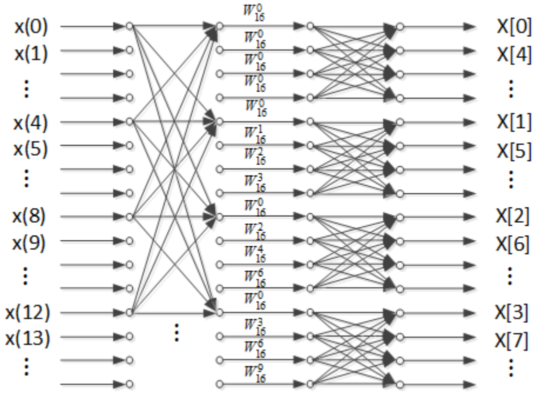


Fig. 5 Dataflow of radix-four FFT

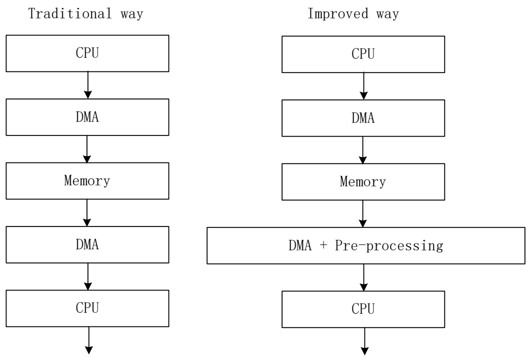


Fig. 6 Comparison of two ways

The CPU in figure.6 is an FFT processor we designed previously. The results are as the table 2 shows. All the data is stored in SRAM which is the memory in figure. 6.

Table 2.Experiment results

| Point<br>Radix | 256 | | 512 | | 1024 | |
|---|---|---|---|---|---|---|
| | traditional | improved | traditional | improved | traditional | improved |
| 2 | 768 | 720 | 1536 | 1460 | 3072 | 2919 |
| 4 | 896 | 784 | 1792 | 1613 | 3584 | 3226 |

According to the table 2, the efficient is improved using the proposed DMA. Our DSP designed previously has also been used to evaluate the DMA system. Some benchmarks such as Fir, Matrix and Convolution have been run on the system. The result shows that DSP could be more efficient using the proposed DMA system.

## Conclusions

This paper presents an intelligent DMA system design with data pre-processing function and the PRAM is improved according to the pre-processing. It supports both floating-point and fixed-point operation which is more flexible for CPU when processing the data. In comparison with the traditional DMA, the proposed design reduces the CPU's burden of simple calculations and increases the efficiency of whole operation. Experiential results show that this design can run at 800 MHz, the area and power of the additional arithmetic units for data preprocessing is 0.45 $mm^2$ and 20 mw, respectively.

## References

[1] G. Netzer, L. Johnsson, D. Ahlin, E. Stotzer, P. Varis, and E. Laure, "Exploiting DMA for performance and energy optimized STREAM on a DSP," *Proc. Int. Parallel Distrib. Process. Symp. IPDPS*, pp. 805–814, 2014.

[2] S. Pan, Q. Guan, and S. Xu, "Optimizing Video Processing Algorithm with Multidimensional DMA Based on Multimedia DSP," pp. 362–365, 2010.

[3] L. Mei, H. Lin, and L. Xue, "The Application of MATLAB7.0 in Developing TIC2000 DSP on System Level(in Chinese)," *MICROPROCESSORS*, pp. 167–169, 2009.

[4] Y. Chen, Y. Jin, and X. Jing, "A TI C5410 DSP-based Real Time System of Connected Digit Recognition(in Chinese)," *J. GUILIN Univ. Electron. Technol.*, pp. 5–8, 2003.

[5] G. Yang, X. Gao, L. Wang, and Z. Wang, "Technology of C/C++ Program Optimization Based on TI C6000 DSP."

[6] F. Chen, L. Fei, and C. Duan, "The Application of EDMA in TD-LTE-A System," *Int. Symp. Instrum. Meas. Sens. Netw. Autom.*, pp. 972–974, 2013.

[7] Texas Instruments, "Enhanced Direct Memory Access (EDMA3) Controller User's Guide," no. March, 2008.

[8] P.Wu, Z. Li, "Design of an Enhanced DMA with Data Pre-computing for Digital Image Processing,": submitted to 2016 International conference on computer science and technology.