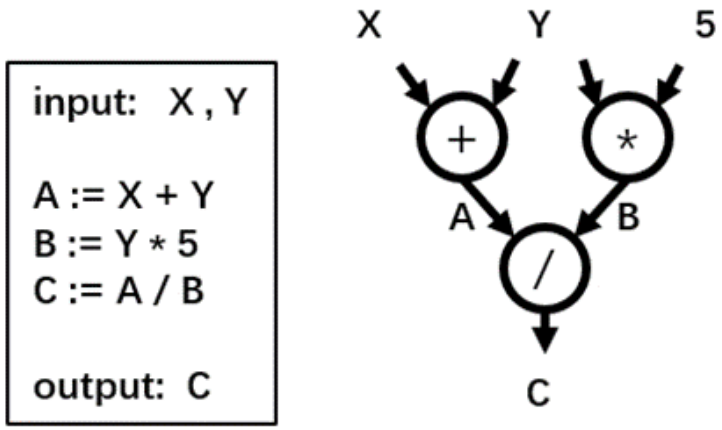
Accelerating the Task Activation and Data Communication for Dataflow Computing

CONTENTS

INTRODUCTION

INTRODUCTION

hybrid dataflow/von-Neumann architectures



The experimental results show that the performance is improved by 3.07%~10.32% under the random data flow graph, the performance of inter-core communication is improved by 4% and the hardware acceleration effect is achieved.

Other works

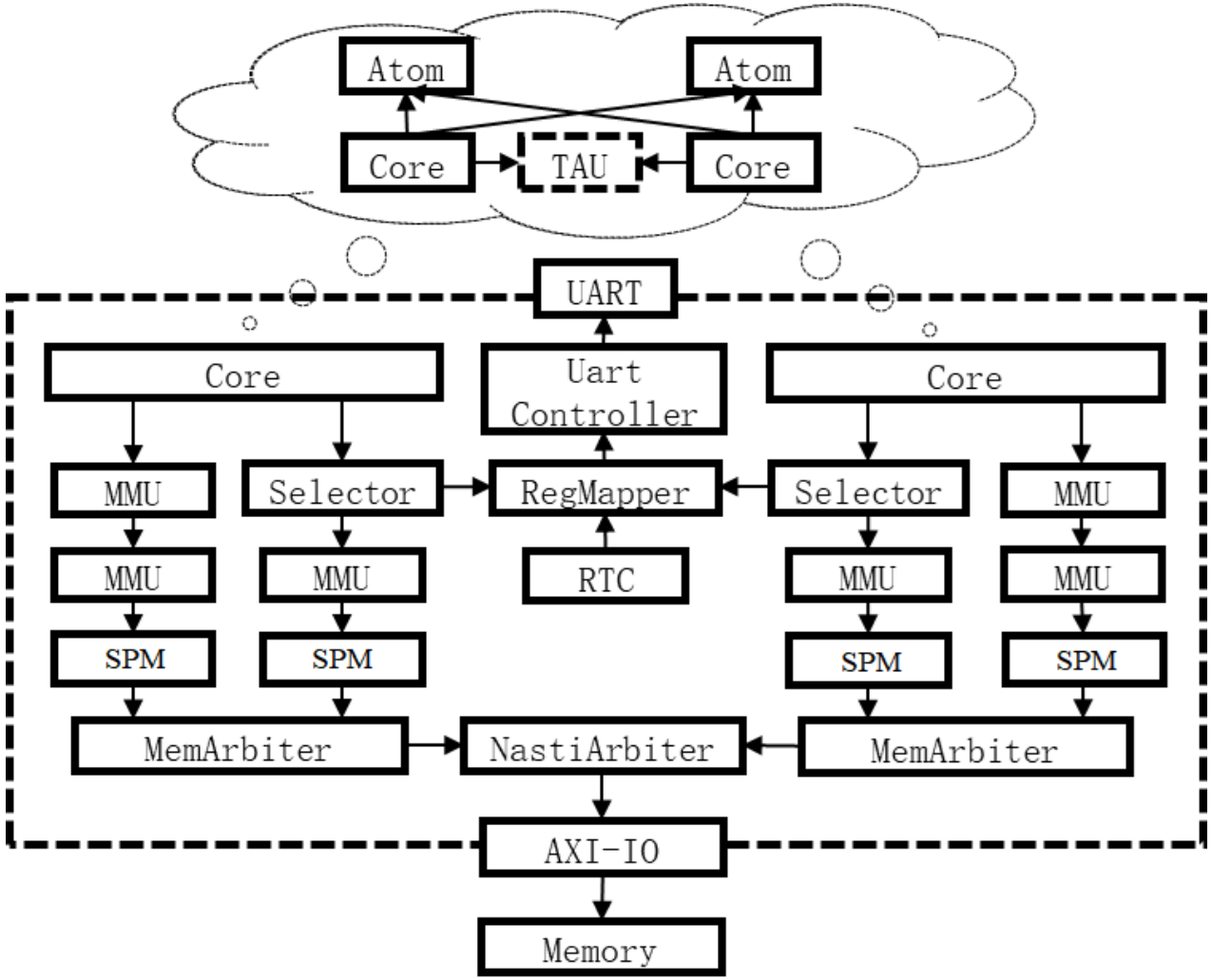
	Task granularity	ISA	Optimize Scene	Generality
stream-dataflow[]	Coarse granularity	SpecificISA	Repeat Pipeline Calculation	Wider
Flexflow[]	Finegranularity	/	CNN Neural Network	narrow
SPU	Instruction Level Fine Granularity	RISC	Loop Iterative Data Flow Calculation	Wider
This Work	Thread Level Coarse Granularity	RISC-V	Multithread-based data flow	wide

System Architecture

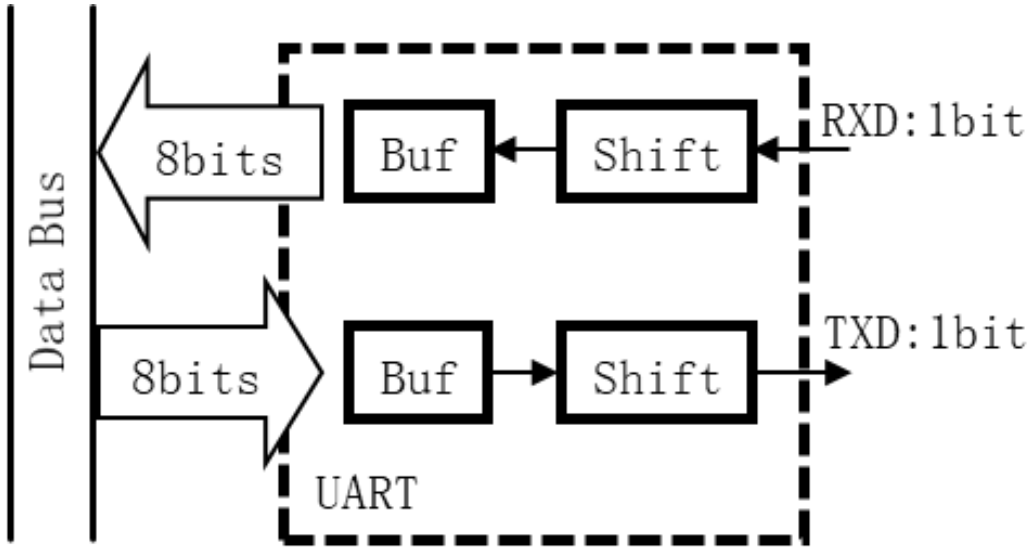
The system architecutre of the machine

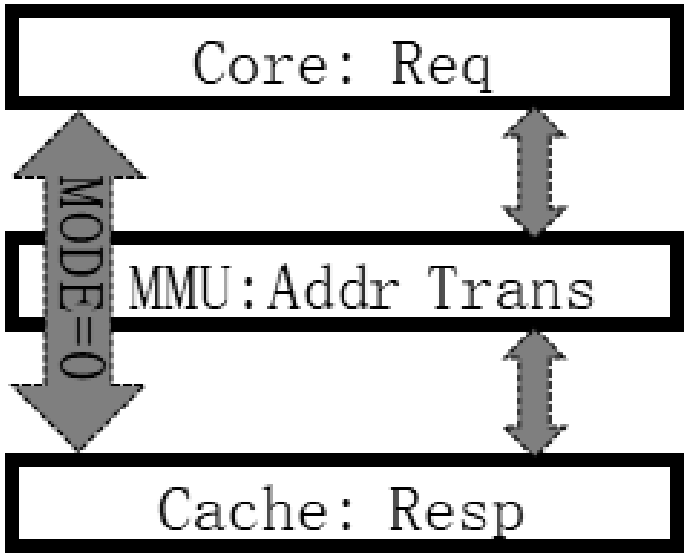
- Chisel
- Verilog
- RISC-V

- RISC-V MINI Core



UART And MMU



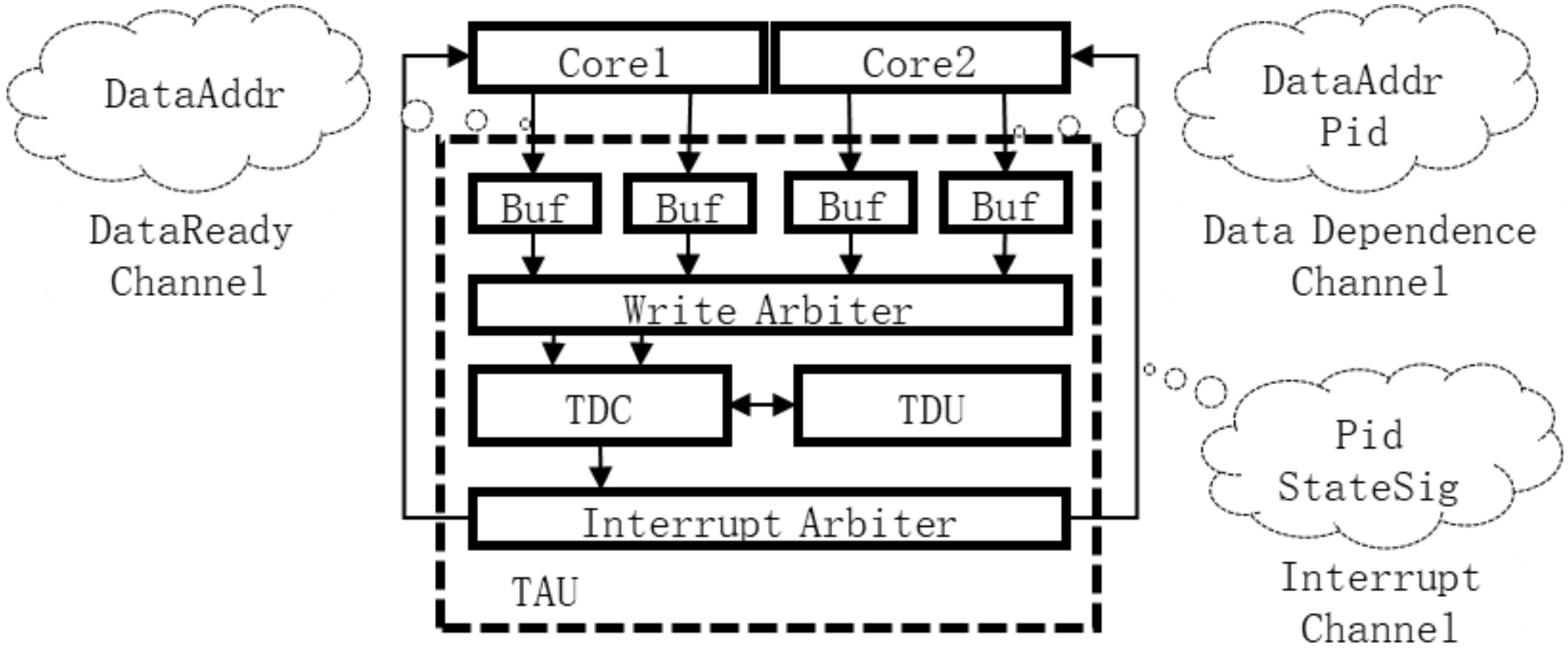


io->UART -> FAT32 .etc->usb-ttl -> computer COM port

TAU Hardware

The basic structure of TAU

Task Activated Unit



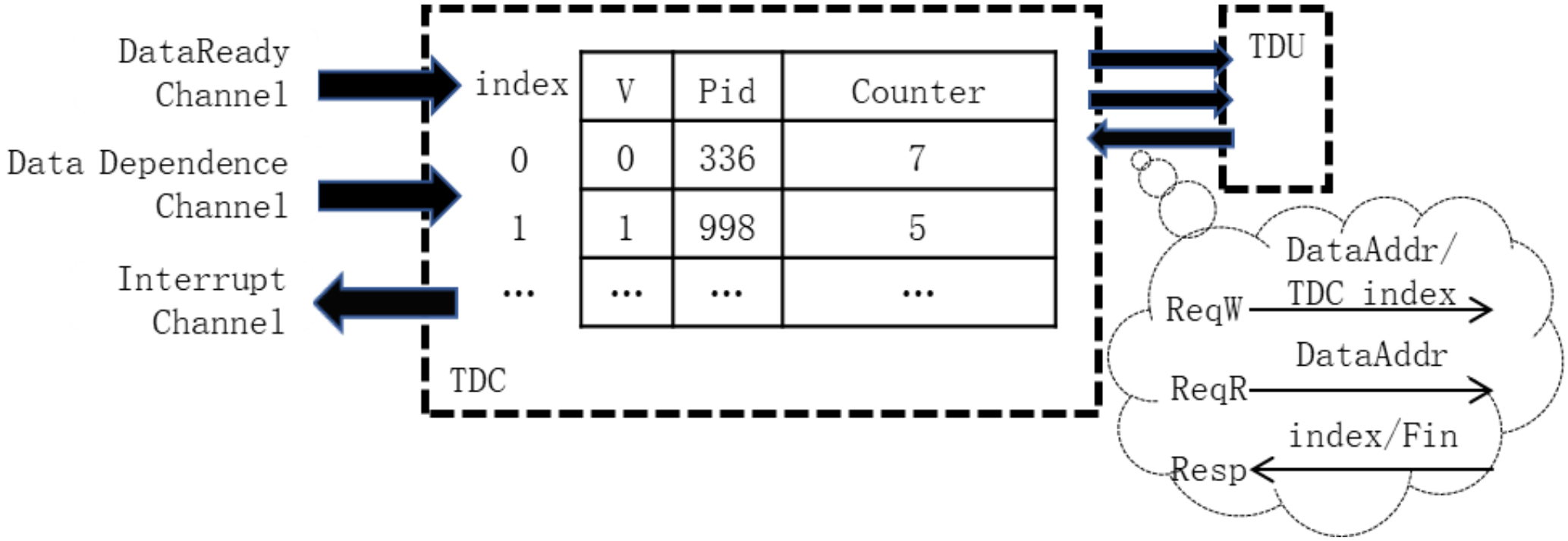
Expansion instruction required by task activation unit

- Based on the RISC-V Instruction Set Manual. Volume 1: User-Level ISA, Version 2.0.

	31 25	24 20	19 15	14 12	11 7	6 0
	Saved	rs2	rs1	Funct3	rd	Opcode
TW	0000000	DataAddr	Pid	000	result	1110100
TR	0000000	DataAddr	00000	000	00000	1110101

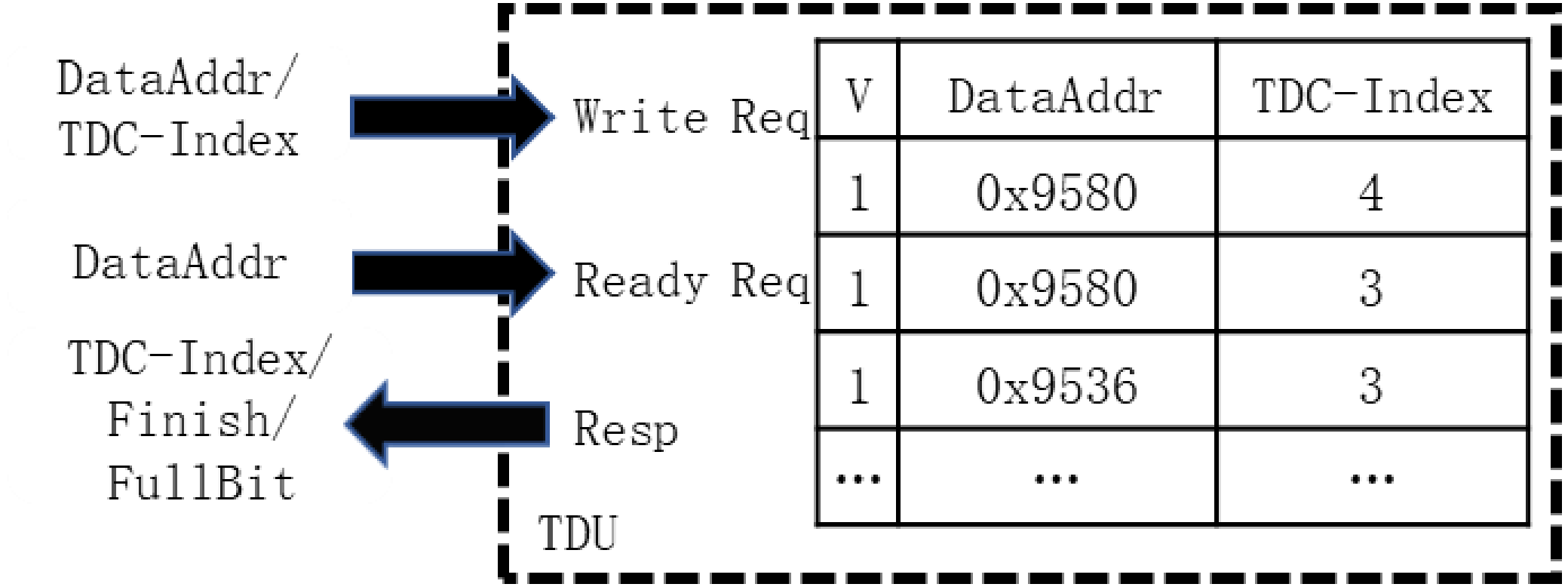
The basic structure of the TDC

- TDC, Task Dependence Counter
- Count the dependent data of the task node, and feed back the interrupt signal, TDC status signal, etc. to the processor core.



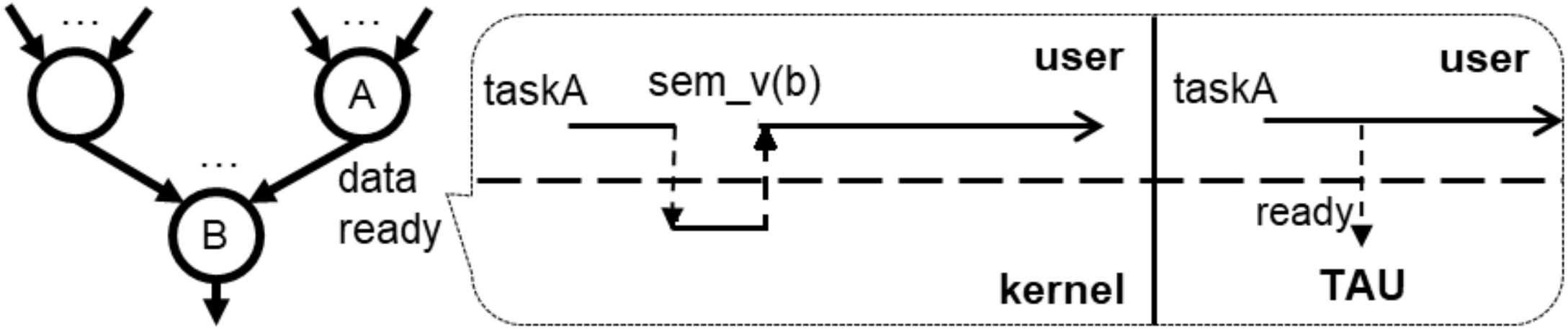
The basic structure of the TDU

- Task-Data dependence Unit， TDU
- Store the dependency between task nodes and data

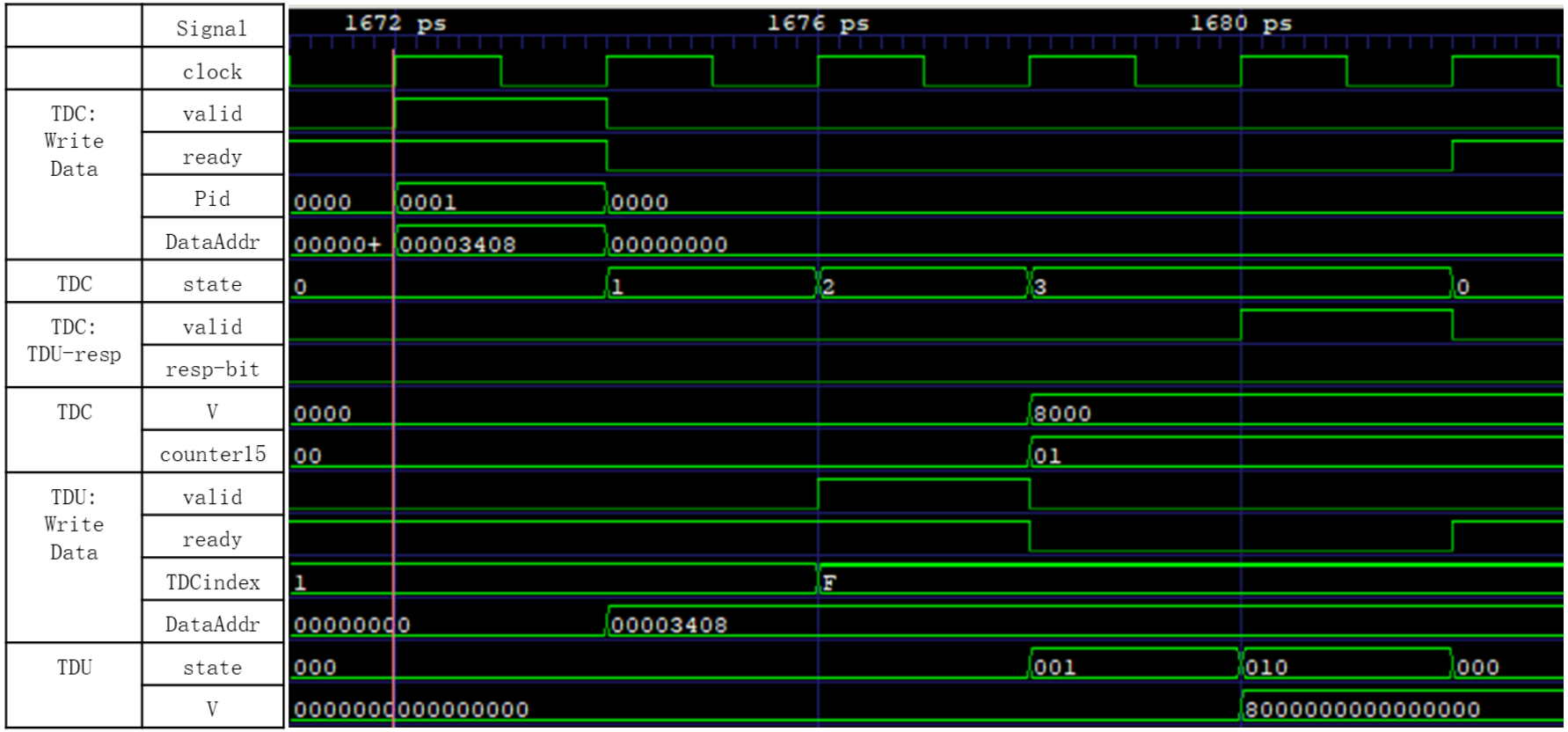


Theoretical acceleration performance analysis of Tau

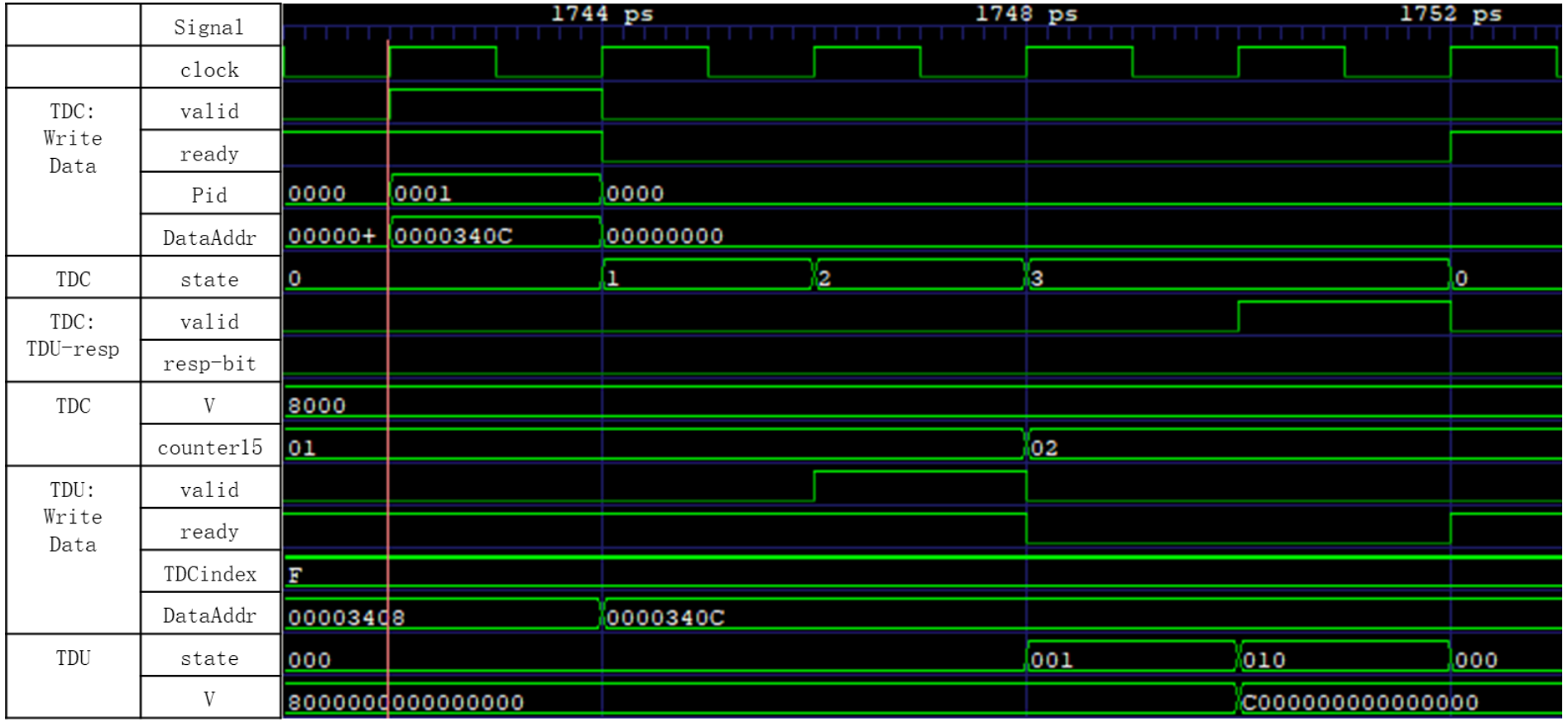
- Trigger mode of data ready signal



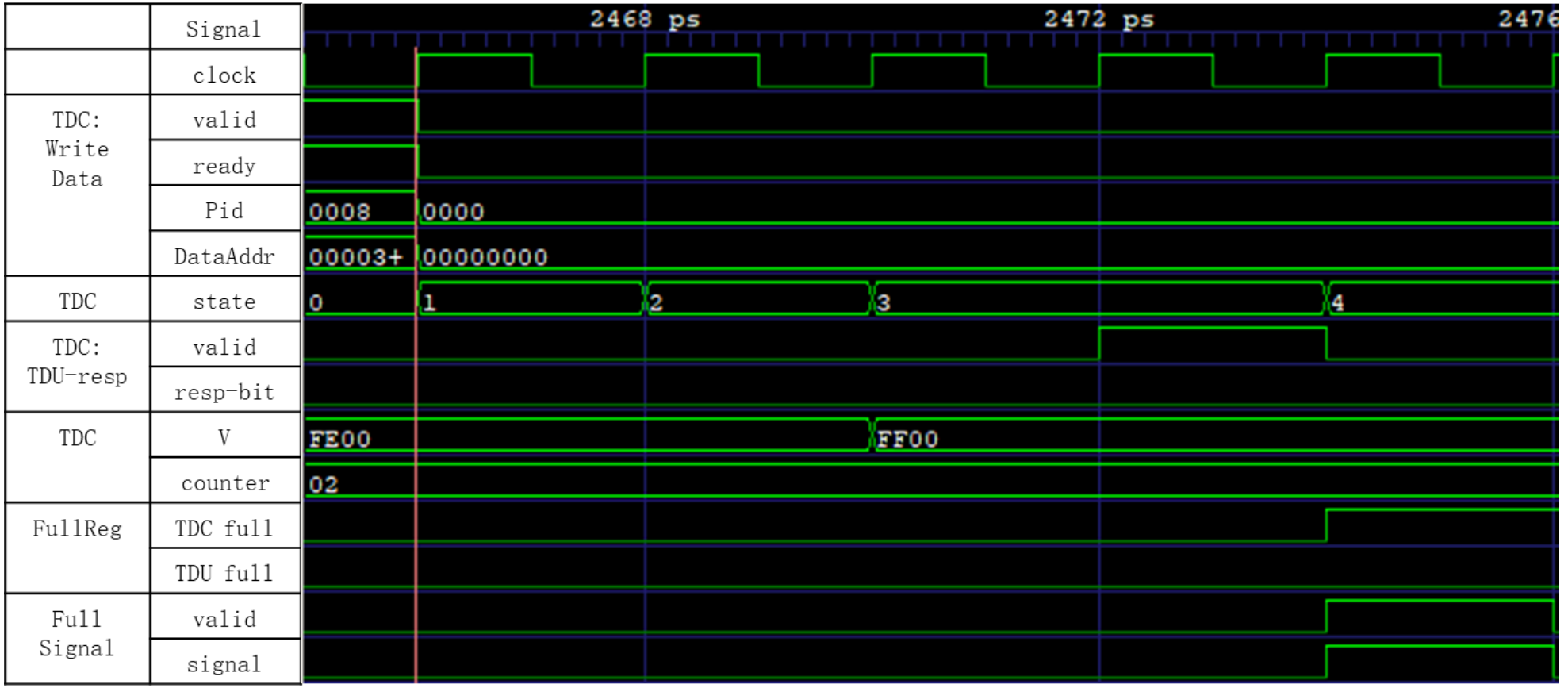
TDC and TDU write simulation



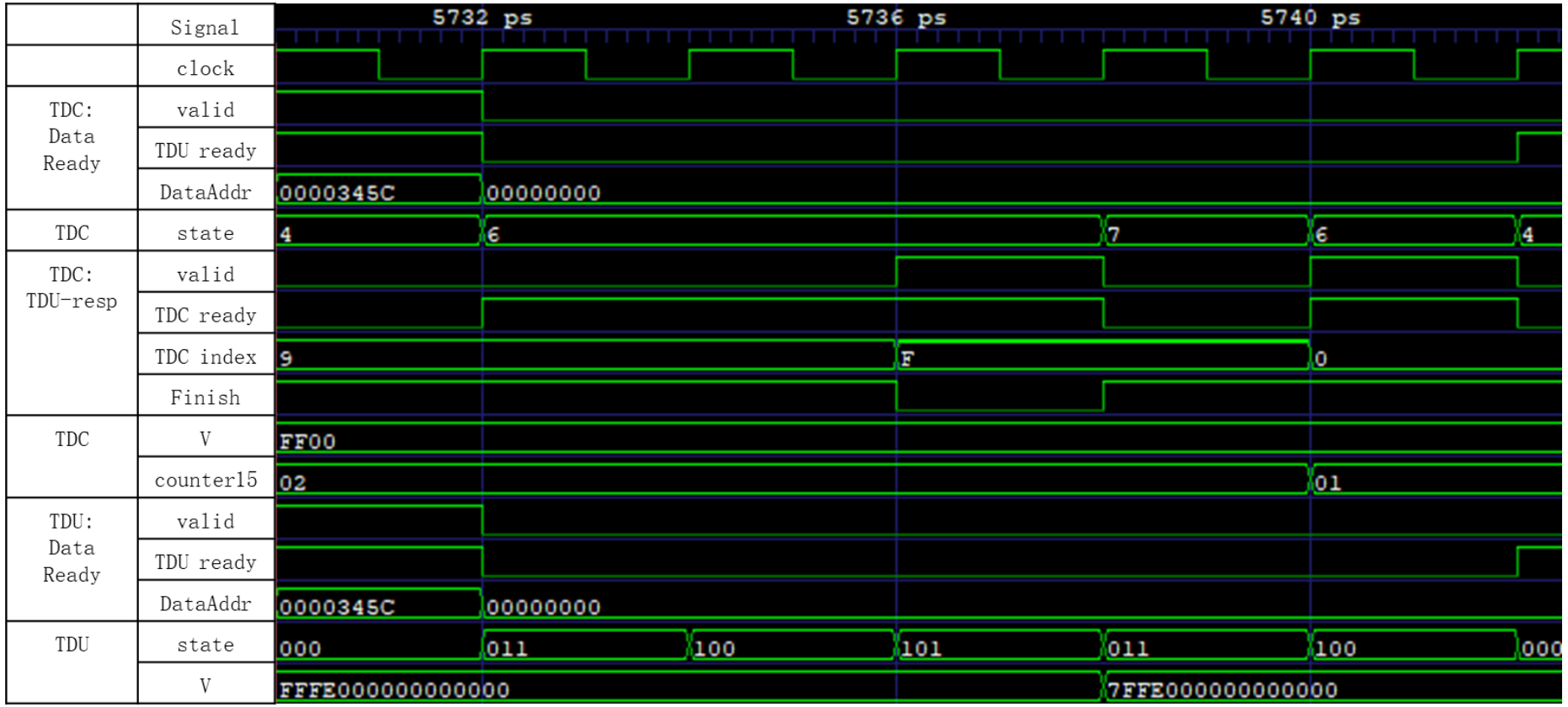
Simulation of data dependent secondary writing of task thread



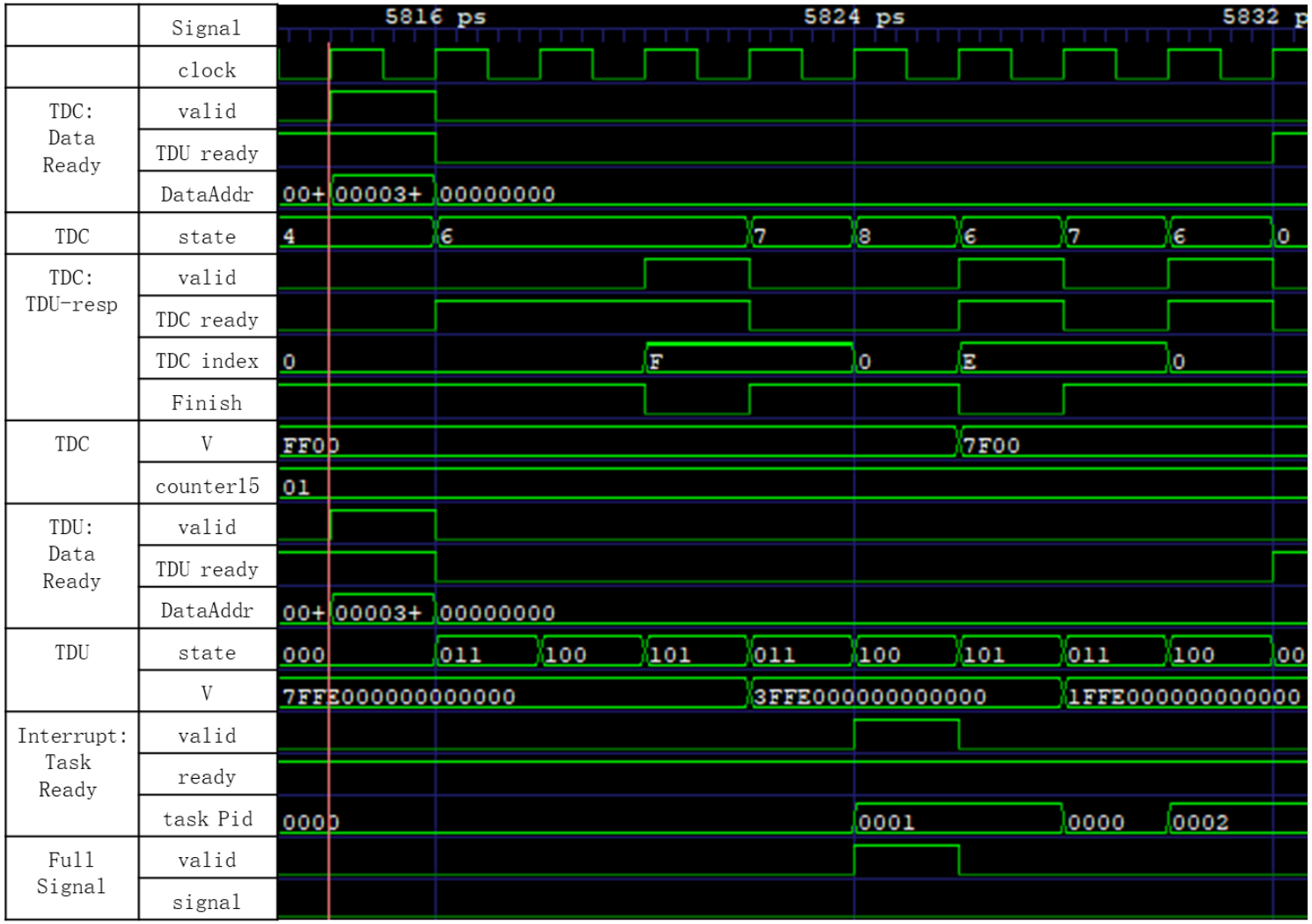
TDC full simulation



Data ready signal trigger count



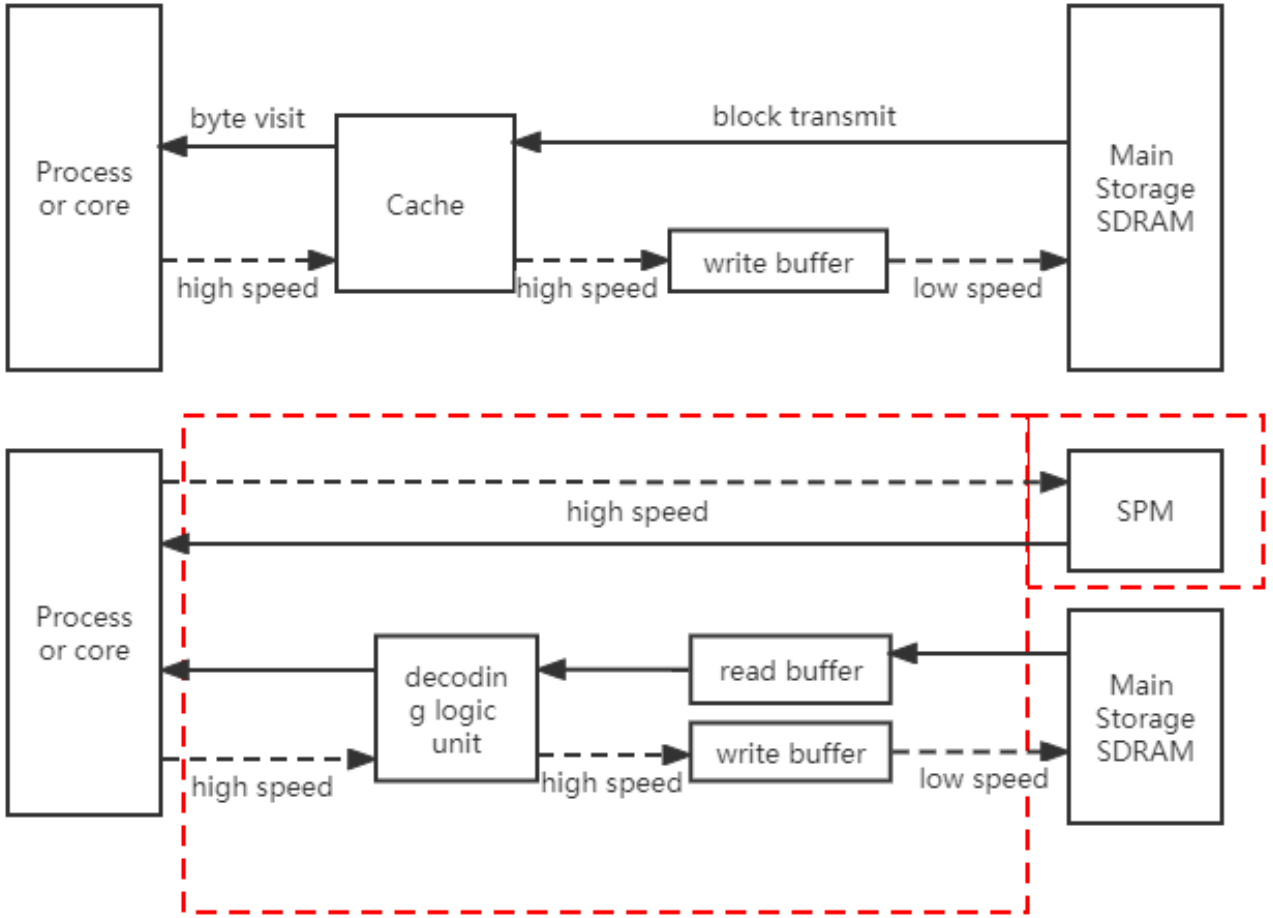
Data ready signal triggers task ready



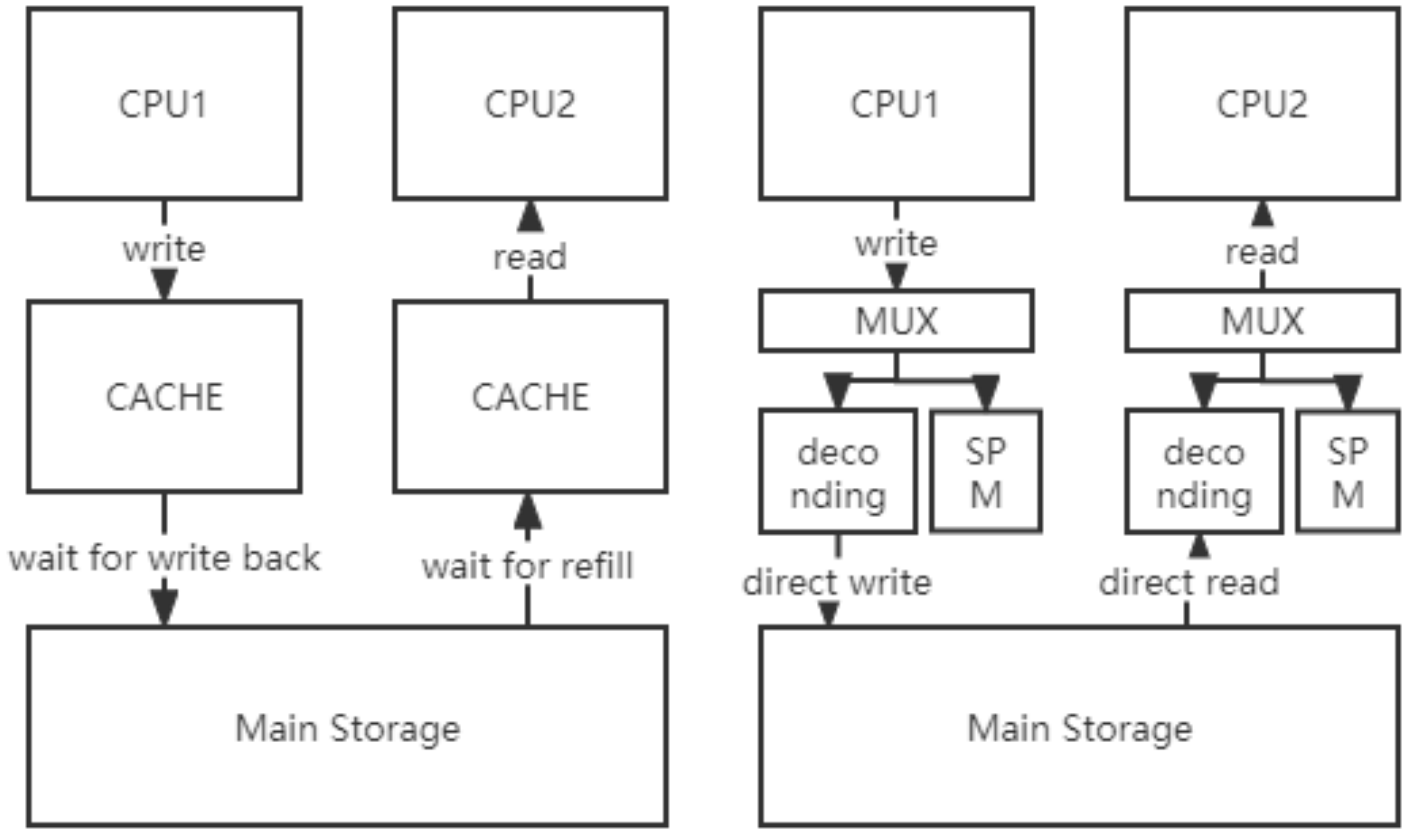
SPM System

The comparison between cache and SPM system

- Cache
- SPM
- Compare



The difference of data flow direction between cache and SPM system



System Performance

Testing Environment

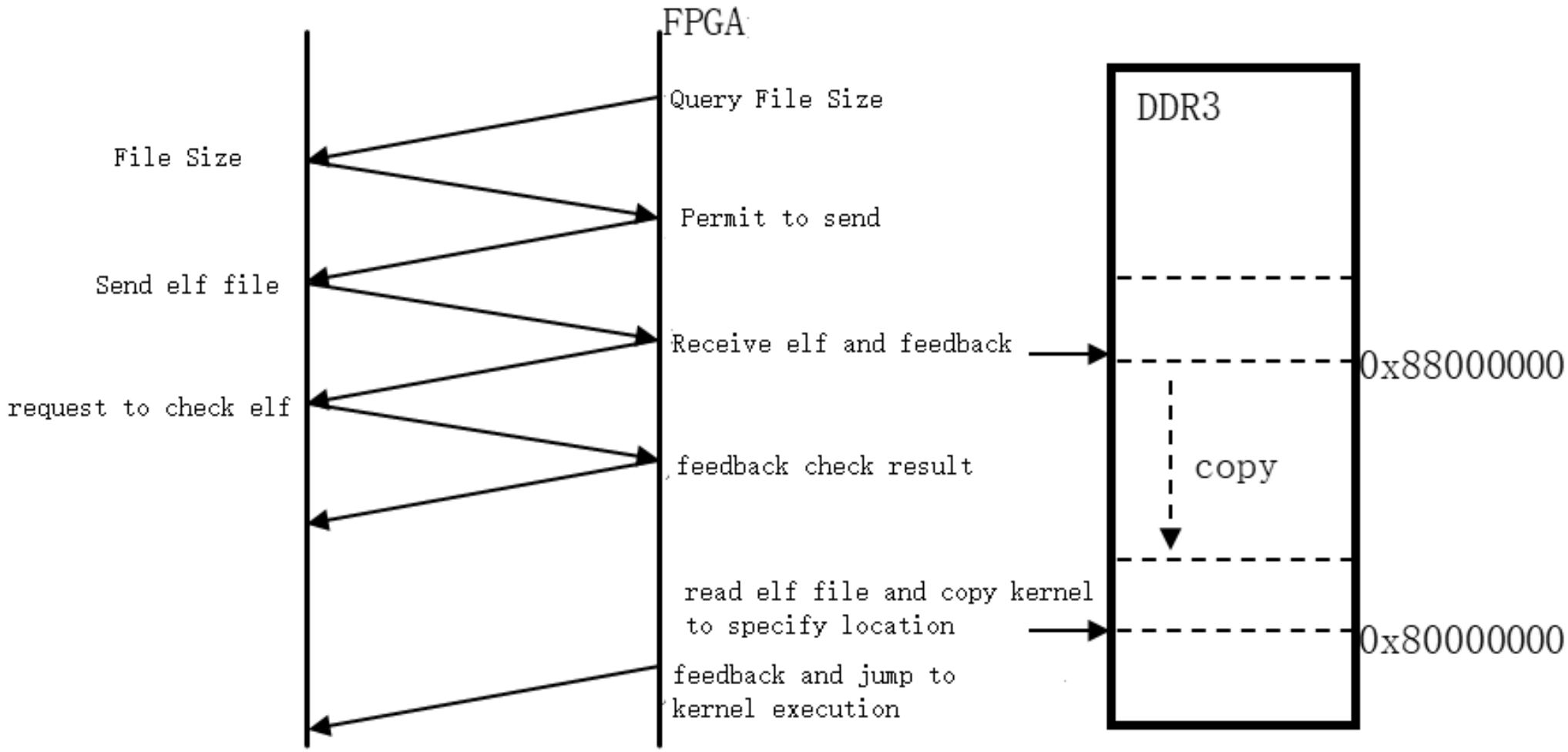
- FPGA Board: Arty A7-100T
- OS: RISCV-XV6
- Graph Tool: TGFF

FPGA Usage

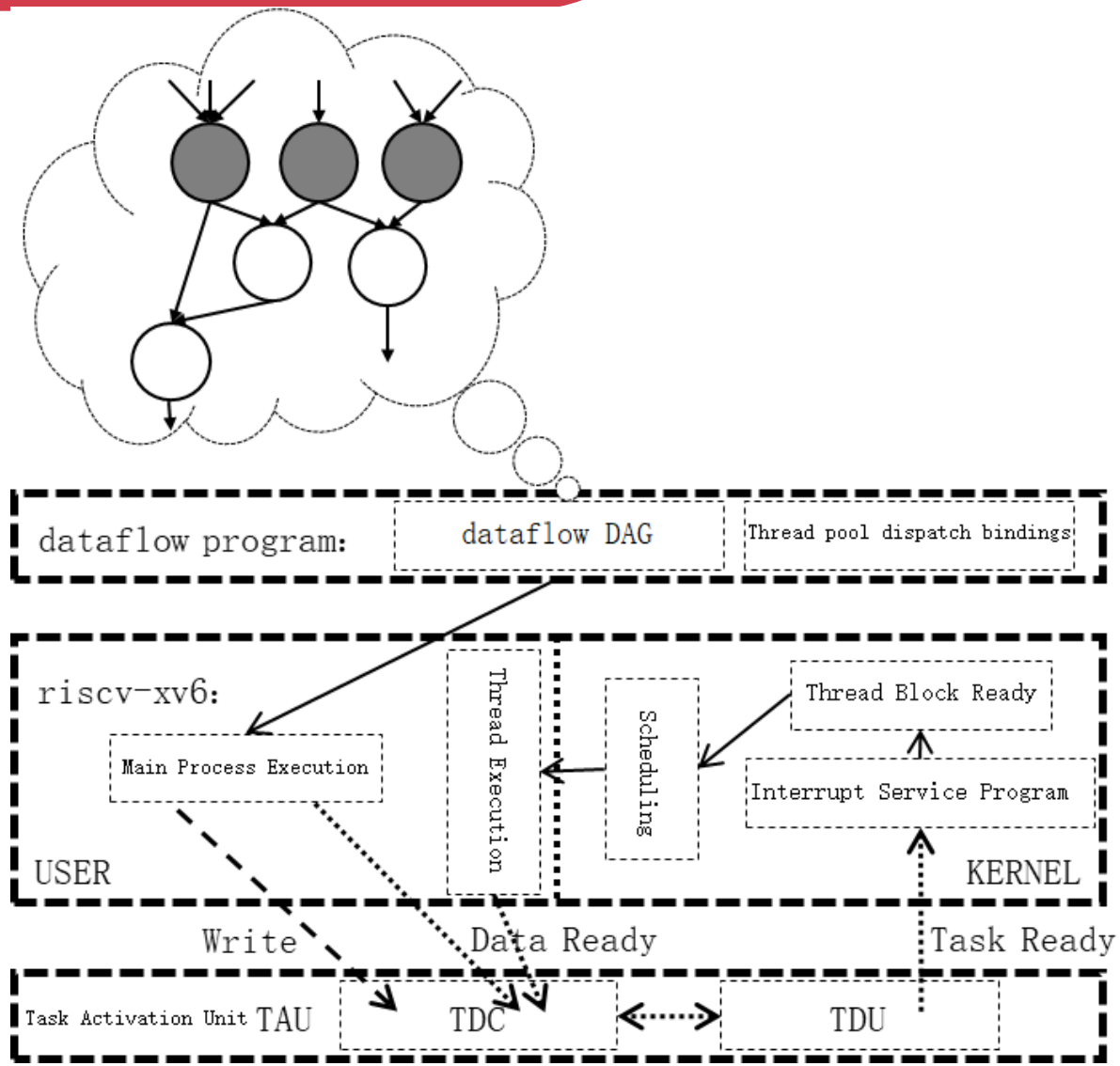
LUTs	LUTRAMs	BRAMs	IOs	FFs
112600	24040	189	262	156353

	RISC-V Processor include TAU		RISC-V Processor Not include TAU	
	Utilization	Utilization%	Utilization	Utilization%
LUT	49200	77.6	22319	35.2
LUTRAM	5040	26.53	4932	25.96
FF	29553	23.31	17372	13.70
BRAM	54	40.00	54	40
IO	52	24.76	52	24.76
BUFG	6	18.75	6	18.75
MMCM	2	33.33	2	33.33
PLL	1	16.67	1	16.67

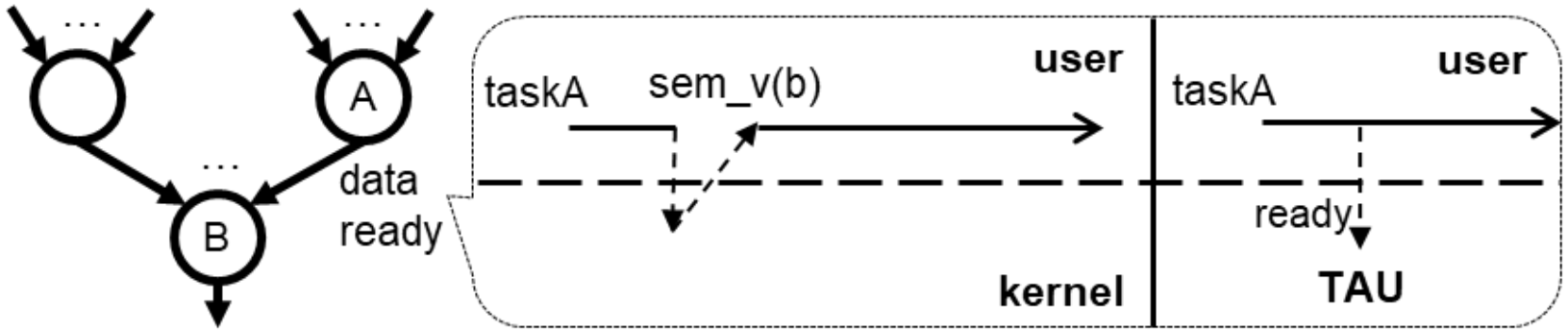
Operating System Loading



Operating Mechanism



Theoretical acceleration performance analysis

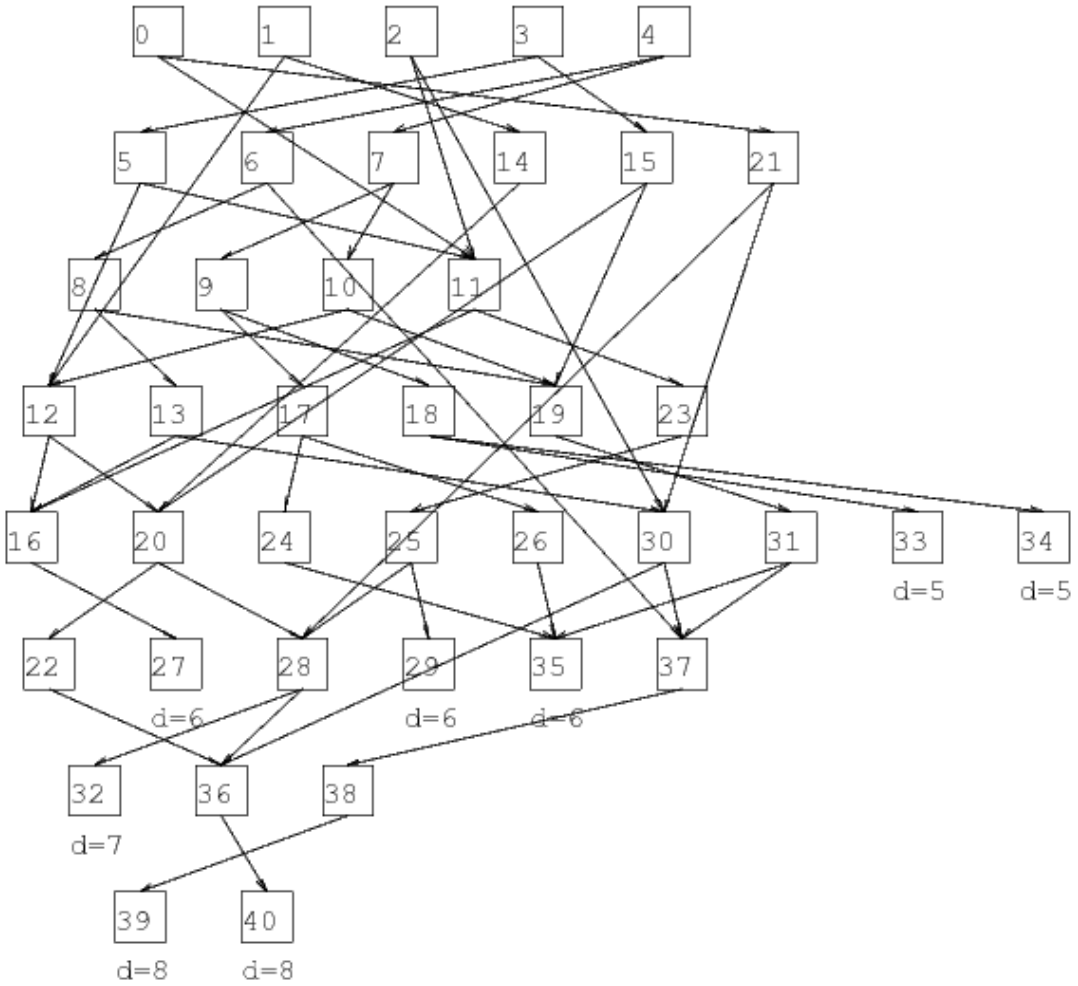


C_i	V_{no}	V_{act}	TW_i	TR_{no}	TR_{act}
1546	1218	22527	5	4	25323

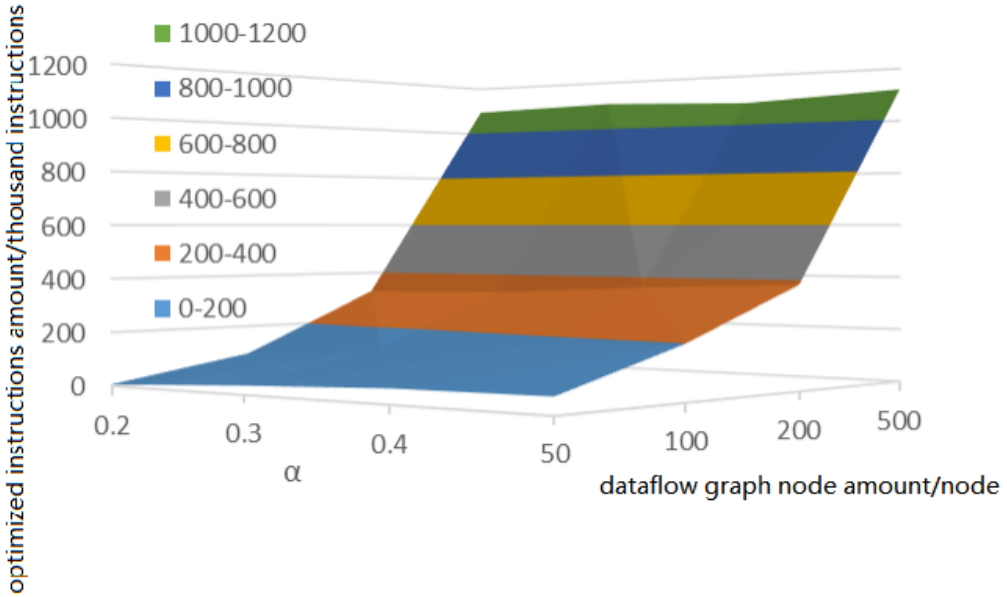
$$Inst_{opt} = \{C_i + (n-1)V_{no} + V_{act}\} - \{nTW_i + (n-1)TR_{no} + TR_{act}\}$$

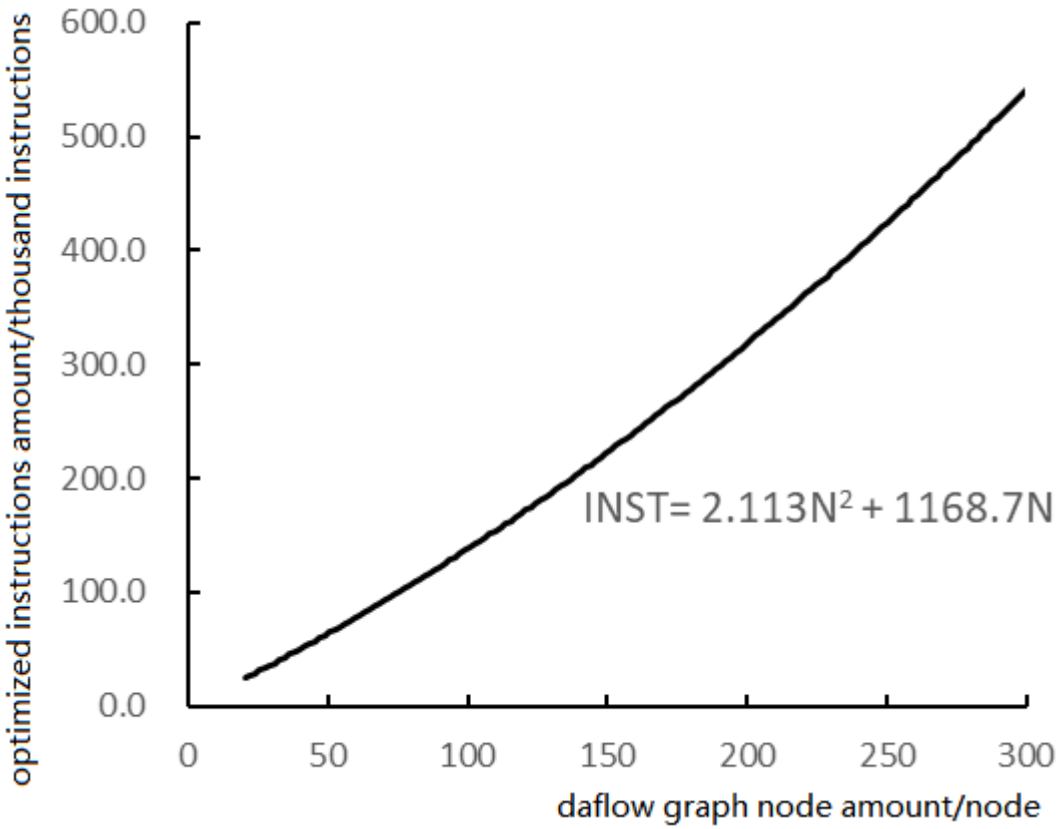
TGFF Configuration

Node Amount	Initial Node Amount	Maximum in-degree	Maximum out-degree	deadline for task execution
N	0.1±0.05N	N	0.2IN	0.5N



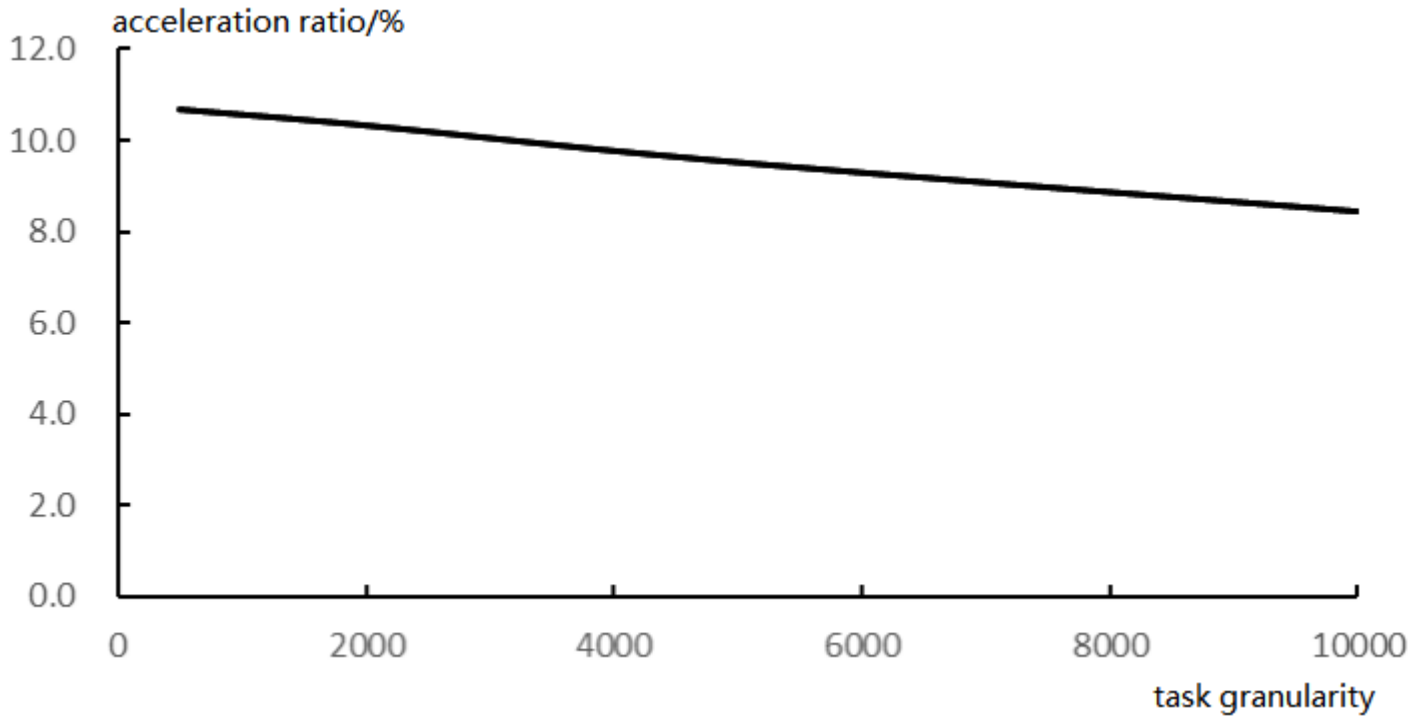
Optimize the relationship between the number of machine instructions and data flow graph nodes, entry parameters





Relationship between TAU Acceleration and Task Granularity

Task granularity	500	2000	5000	10000
TAUacceleration	10.67%	10.32%	9.51%	8.42%



The compute of the acceleration ratio

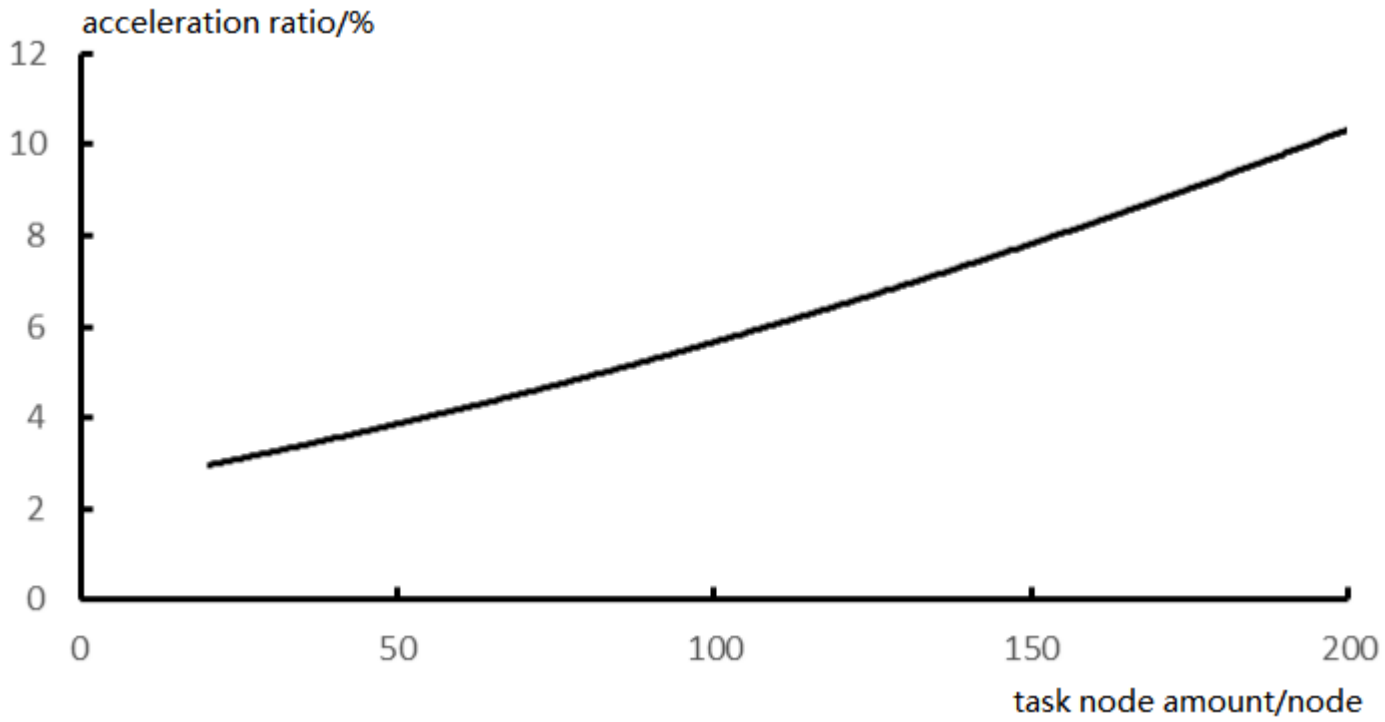
a.Adopt semaphore synchronization
b.Adopt data flow hardware TAU

Record the operation totally used clock cycle count of both: Ta,Tb.

Acceleration ratio = (Ta-Tb)/ Ta x 100%

Relationship between TAU Acceleration and Number of Task Nodes

Task Amount	20	50	100	200
TAUacceleration	3.07%	3.63%	5.80%	10.32%



CONCLUSION

The work of this paper design and implement a new kind of hybrid dataflow/von-Neumann machine.

1.Design special hardware for data flow, namely task activation unit TAU. It implements the functions of writing the task node dependent data relationship, triggering the dependent data ready signal, and triggering the task node ready signal. At the same time, the write instruction and ready instruction are extended to access the unit, and the programming verification is carried out in the way of assembly embedded.

2.Integrate SPM system into hybrid machine to enhance the efficiency of inter-core communication. It implements the functions of maintain the coherency of the main storage and storing special data in SPM unit.

3.System performance analysis of data flow execution environment. On the one hand, the random data flow graph generated by TGFF tool is used to help describe the accelerated optimization of tau, and the mathematical relationship between the number of nodes in the data flow graph and the number of optimized instructions is established. On the other hand, the overall execution environment of data flow is simulated, the acceleration ratio of running time is given, and the acceleration performance is discussed.

ACKNOWLEDGMENTS

This work was supported by teachers and students of SKT group in NHPCC (National High Performance Compute Center). This research was funded by project granted from Shenzhen Science Technology Foundation (20200811201536001).

Thank you for listening!