An RTOS in Hardware for Energy Efficient Software-based TCP/IP Processing

Naotaka Maruyama
Kernelon Silicon Inc.

Tohru Ishihara
Kyushu University

Hiroto Yasuura
Kyushu University

Rev.1.0

KERNELON SILICON

Issues to realize high performance TCP/IP

TCP/IP penetrates not only PCs but also embedded systems

TCP/IP performance in embedded systems is very low.
TCP/IP performance of ARM9 is only 11Mbps at 50MHz clock.

Conventional Solutions

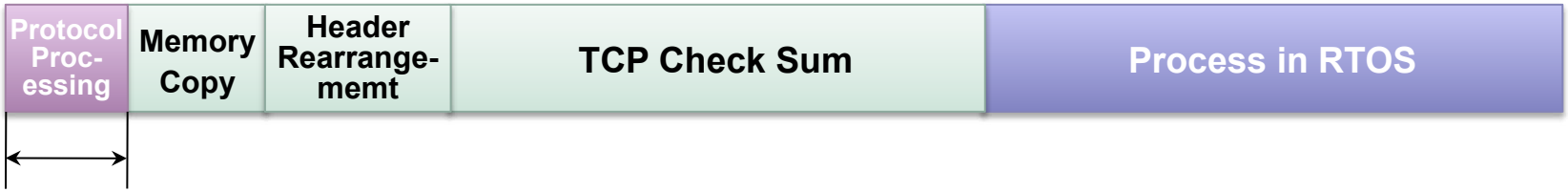
1. **Improve CPU performance (use higher clock rate)**
   - **> Increase power consumption**
   - **> Max TCP/IP throughput depends on upper limitation of clock rate of the CPU**
2. **TCP/IP in hardware (TOE)**
   - **> Disable modify**
   - **> Disable add some functions**
   - **> Bottle neck other than TCP/IP**

KERNELON SILICON

Analysis of TCP/IP Processing

Consumption time for each process during TCP/IP operation

Conventional Firmware TCP/IP



Protocol Processing occupies only about 10% CPU power.

	Conventional TCP/IP
Protocol Processing	10.5%
Header Rearrangement	15.3%
Check Sum	32.2%
Memory Copy	9.9%
RTOS	32.2%
	100.0%

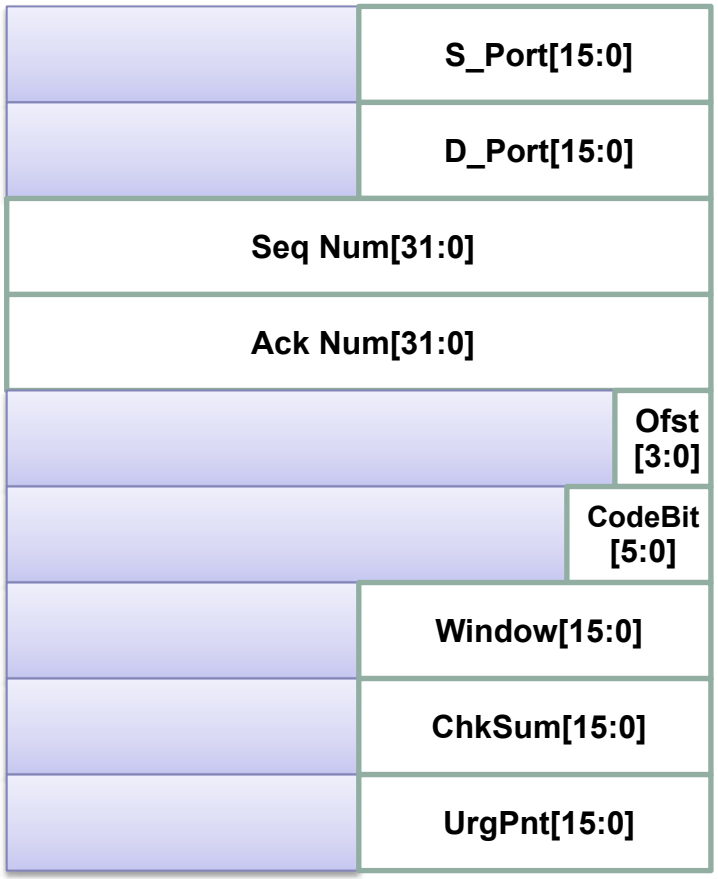
KERNELON SILICON

Header Rearrangement

Buffer Memory

D_Port [7:0]	D_Port [15:8]	S_Port [7:0]	S_Port [15:8]
Seq Num [7:0]	Seq Num [15:8]	Seq Num [23:16]	Seq Num [31:24]
Ack Num [7:0]	Ack Num [15:8]	Ack Num [23:16]	Ack Num [31:24]
Window [7:0]	Window [15:8]	CodeBit [5:0]	Ofst [3:0]
UrgPnt [7:0]	UrgPnt [15:8]	ChkSum [7:0]	ChkSum [15:8]

Data Memory

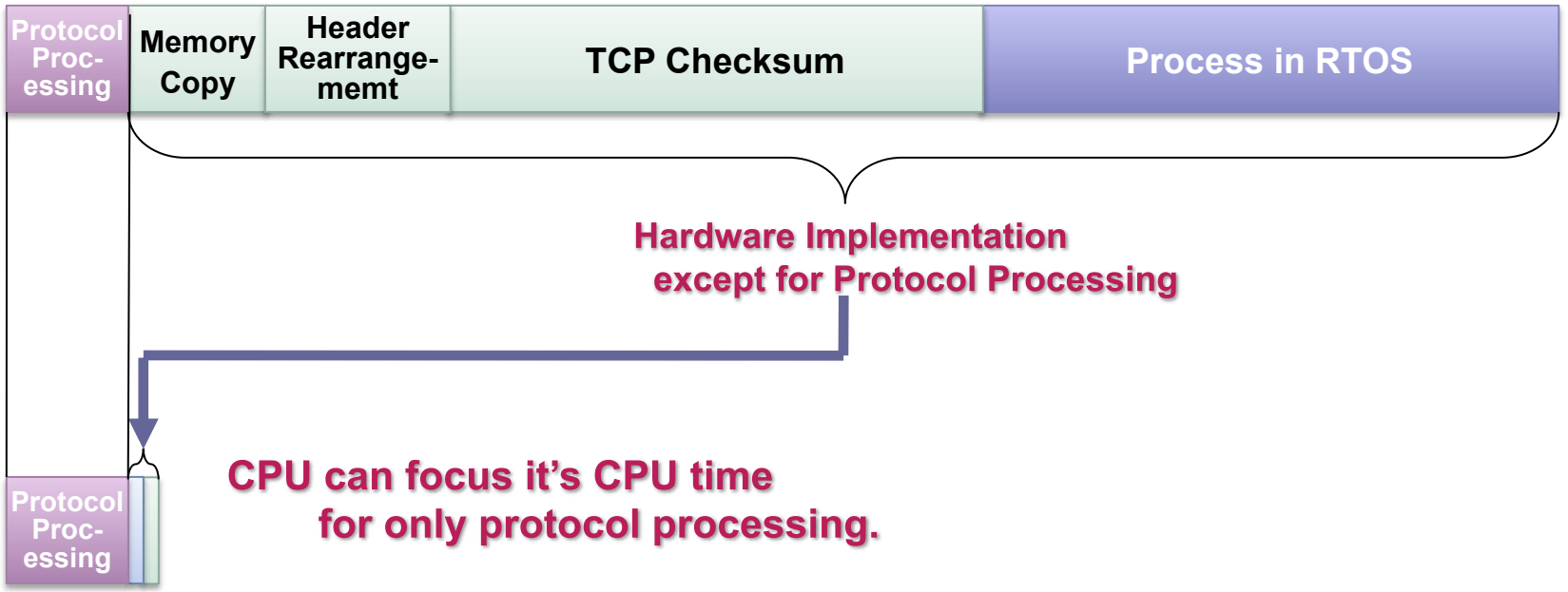


KERNELON SILICON

Analysis of TCP/IP Processing

Consumption time for each process during TCP/IP operation

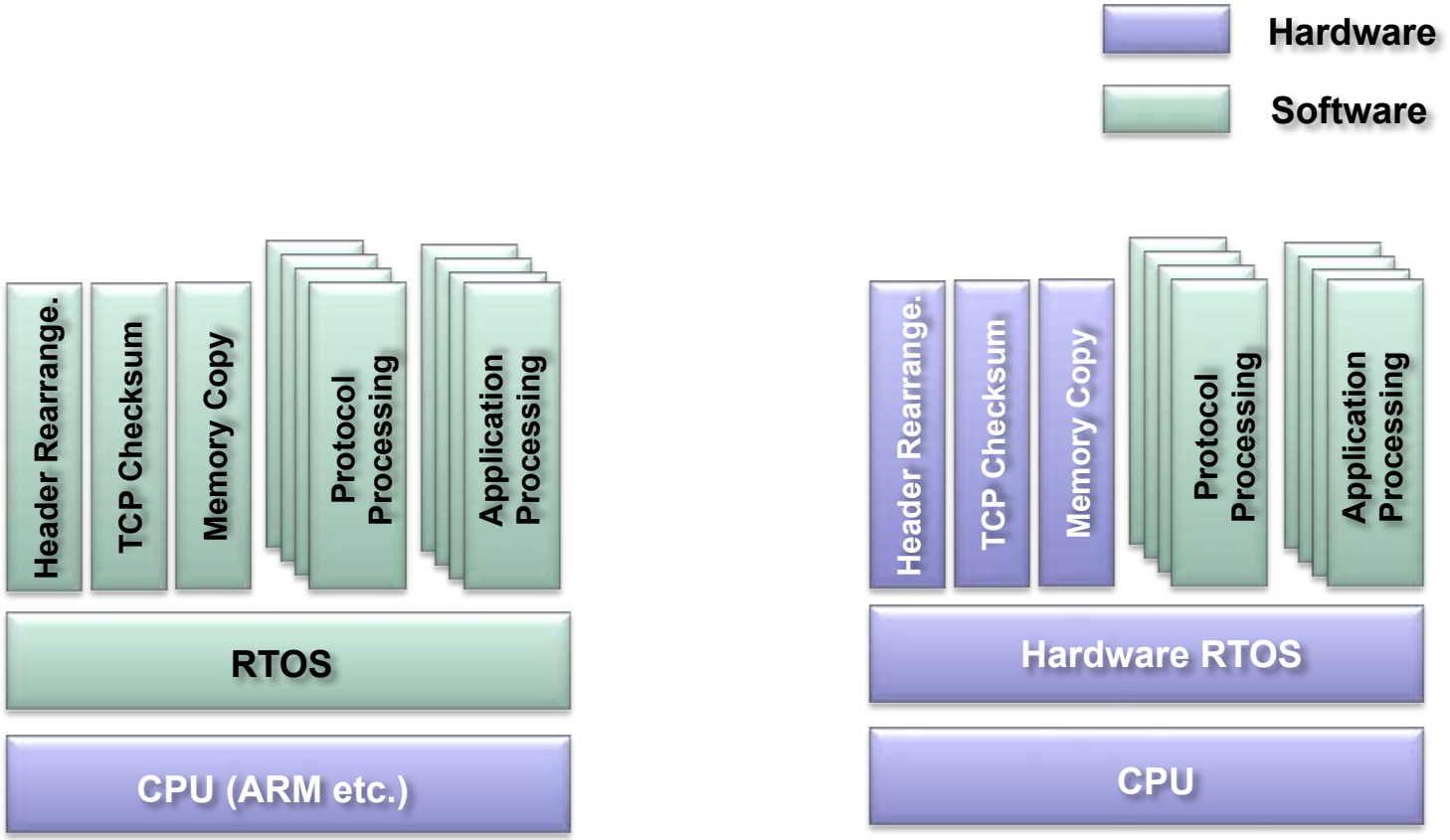
Conventional Firmware TCP/IP



ARTESSO

KERNELON SILICON

Our Approach



Conventional Approach

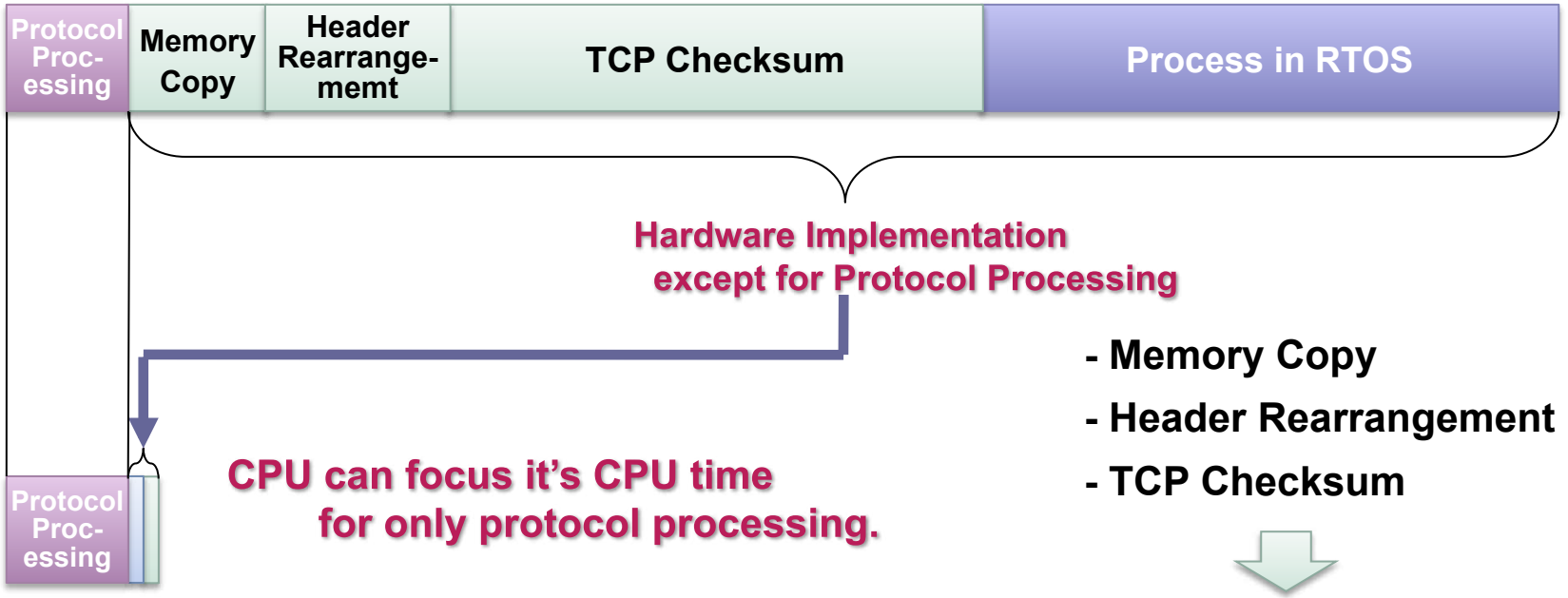
ARTESSO

: Advanced Real Time Embedded Silicon System Operator

KERNELON SILICON

Analysis of TCP/IP Processing

Conventional Firmware TCP/IP



ARTESSO

Easy to implement in hardware

Implement RTOS in hardware has a lot of issues

KERNELON SILICON

Issues in hardware RTOS

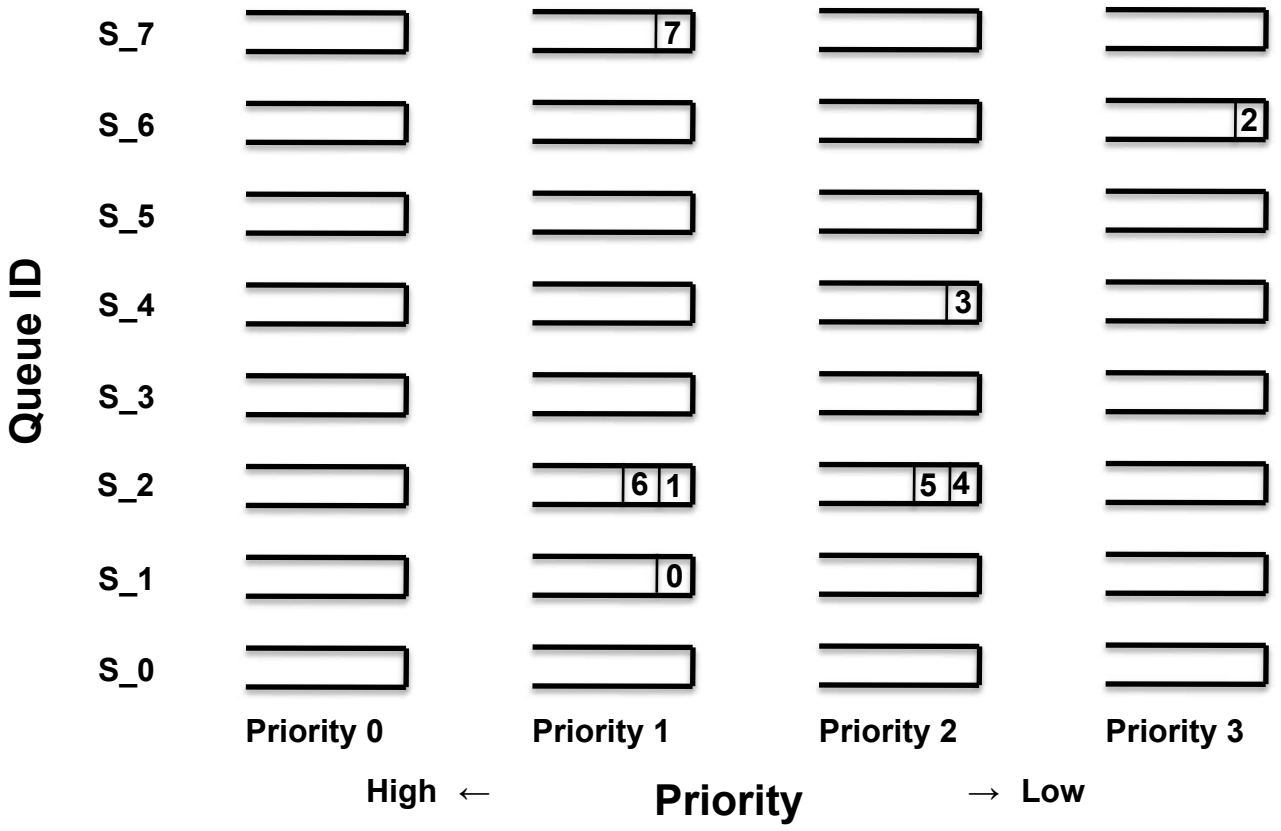
1. RTOS needs a large number of queues
   - Many wait queues are needed for RTOS
   - Example
     - 32 semaphore IDs, 32 Event IDs and 192 mailbox IDs
     - 16 priorities
     - Total number of queues are 4,096
   - Typical hardware FIFO needs great deal of gates.
2. Function of “search and retrieve a task from a wait queue” is needed
   - Execution of the function spends many CPU cycles.
   - Typical hardware FIFO can not provide such a function

We solved those issues by new idea, Virtual Queue.

Queue System using hardware FIFO

A simple example of waiting queue for semaphore

Number of semaphore IDs : 8、Priorities : 4、Number of tasks : 8



Issues

1. Number of Queues = Number of Semaphore IDs x Priorities

2. Each queue needs capacity which can save all tasks

KERNELON SILICON

Virtual Queue

Queue Control Registers keep following information
- Which Queue ID is the task belonging now
- What is the task priority
- What is the order number

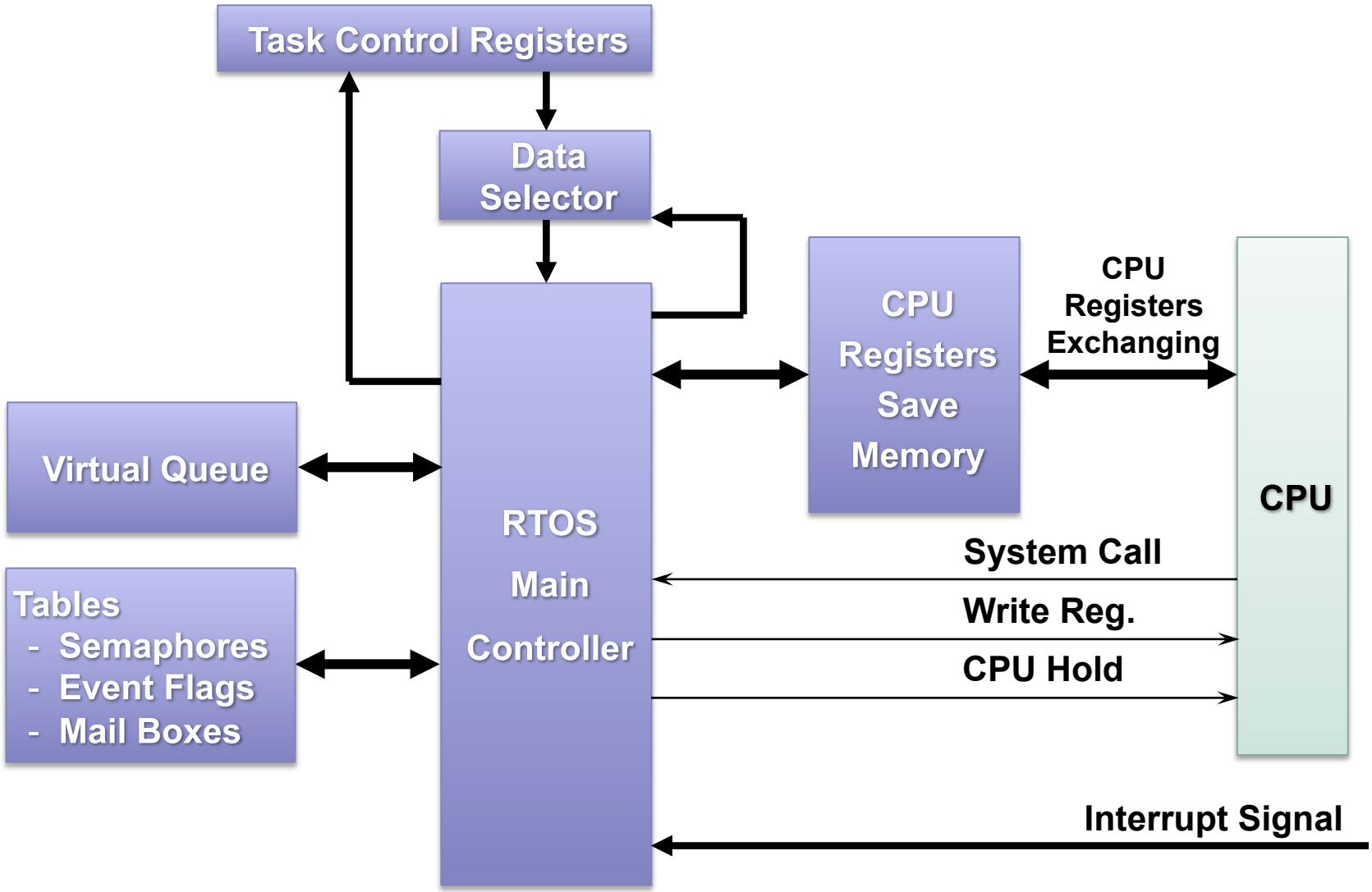
Queue Control Registers

	Task 0 Register	Task 1 Register	Task 2 Register	Task 3 Register	Task 4 Register	Task 5 Register	Task 6 Register	Task 7 Register
Task ID	0	1	2	3	4	5	6	7
Queue ID	S_1	S_2	S_6	S_4	S_2	S_2	S_2	S_7
Priority	1	1	3	2	2	2	1	2
Order	1	6	4	0	7	3	5	2

- Virtual Queue ····· Compress Queue Information
  - Conventional Queue ····· Keeps task IDs in each queue
  - Virtual Queue ····· Manages queue information in each task

KERNELON SILICON

Architecture of hardware RTOS



KERNELON SILICON

ARTESSO RTOS

- Implement RTOS process in hardware
  - System call processing
  - Dispatch processing
- Provide 30 system calls which is used commonly
- Maximum Spec.
  - more than 256 tasks, Event IDs, Semaphore IDs and Mailbox IDs
- All wait queues and ready queues are FCFS.

KERNELON SILICON

System Calls

- Synchronization and communications
  - Create Event Flag
  - Delete Event Flag
  - Wait Event Flag
  - Set Event Flag
  - Poll Event Flag
  - Clear Event Flag
  - Create Semaphore
  - Delete Semaphore
  - Obtain Semaphore
  - Release Semaphore
  - Create Mailbox
  - Delete Mailbox
  - Send to Mailbox
  - Receive from Mailbox
- Task Management
  - Exit Current Task
  - Start Task
  - Terminate Task
  - Change Priority
- System Status Management
  - Lock CPU Lock
  - Unlock CPU Lock
  - Disable Dispatch
  - Enable Dispatch
  - Get the Running Task ID
  - Rotate Task Priority Order
- Time Management
  - Set System Time
  - Refer to System Time
- Synchronous Function with Task
  - Wakeup Task
  - Sleep Task
  - Release Waiting Task

Evaluation

KERNELON SILICON

RTOS Performance Comparison

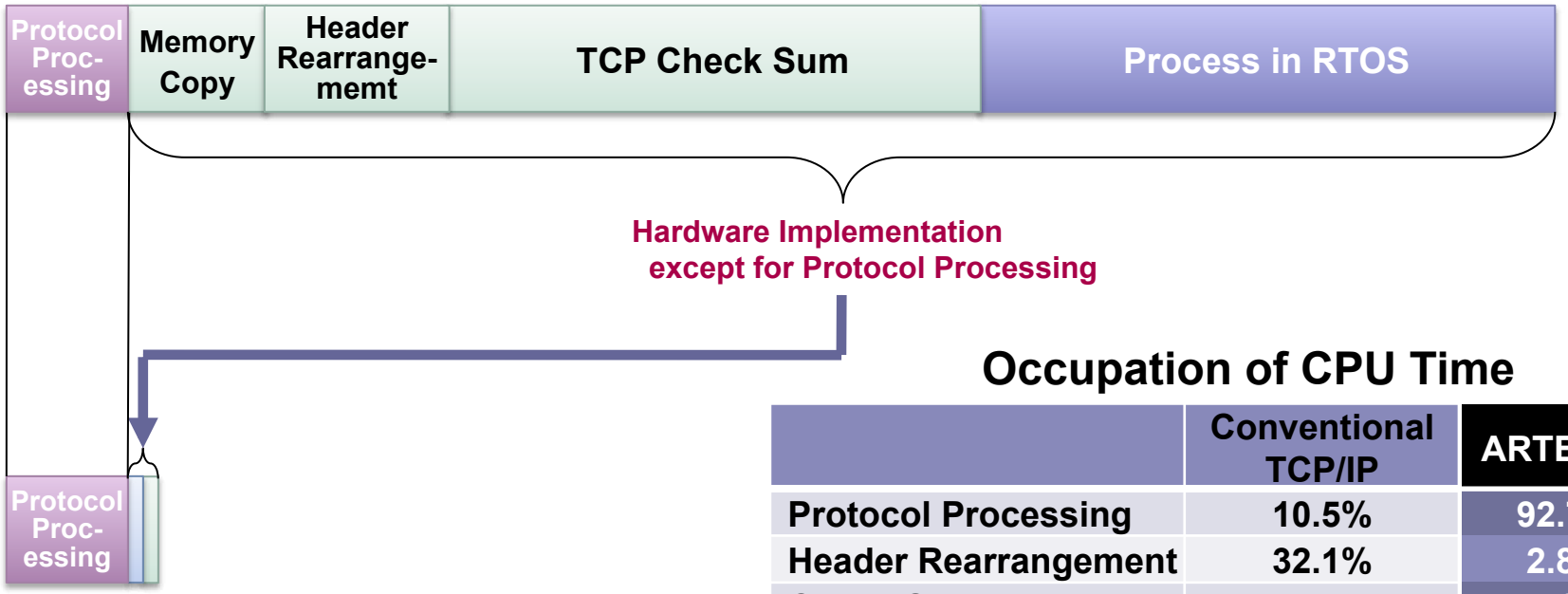
Clock Counts to needed for each system call

System Call	Dispatch	NORTi (ARM926)	ARTESSO V4.2
Sleep Task	Yes	628	10
Wakeup Task	Yes	496	10
Receive from Mailbox	No	224	7
Receive from Mailbox	Yes	591	11
Send to Mailbox	No	360	8
Send to Mailbox	Yes	541	11
Obtain Semaphore	No	216	6
Obtain Semaphore	Yes	558	9
Release Semaphore	No	344	7
Release Semaphore	Yes	536	11

NORTi: A RTOS conforms to ITORN
ITRON : A Standard of RTOS in Japan, very similar to NUCLEOUS

KERNELON SILICON

Result of TCP/IP Processing



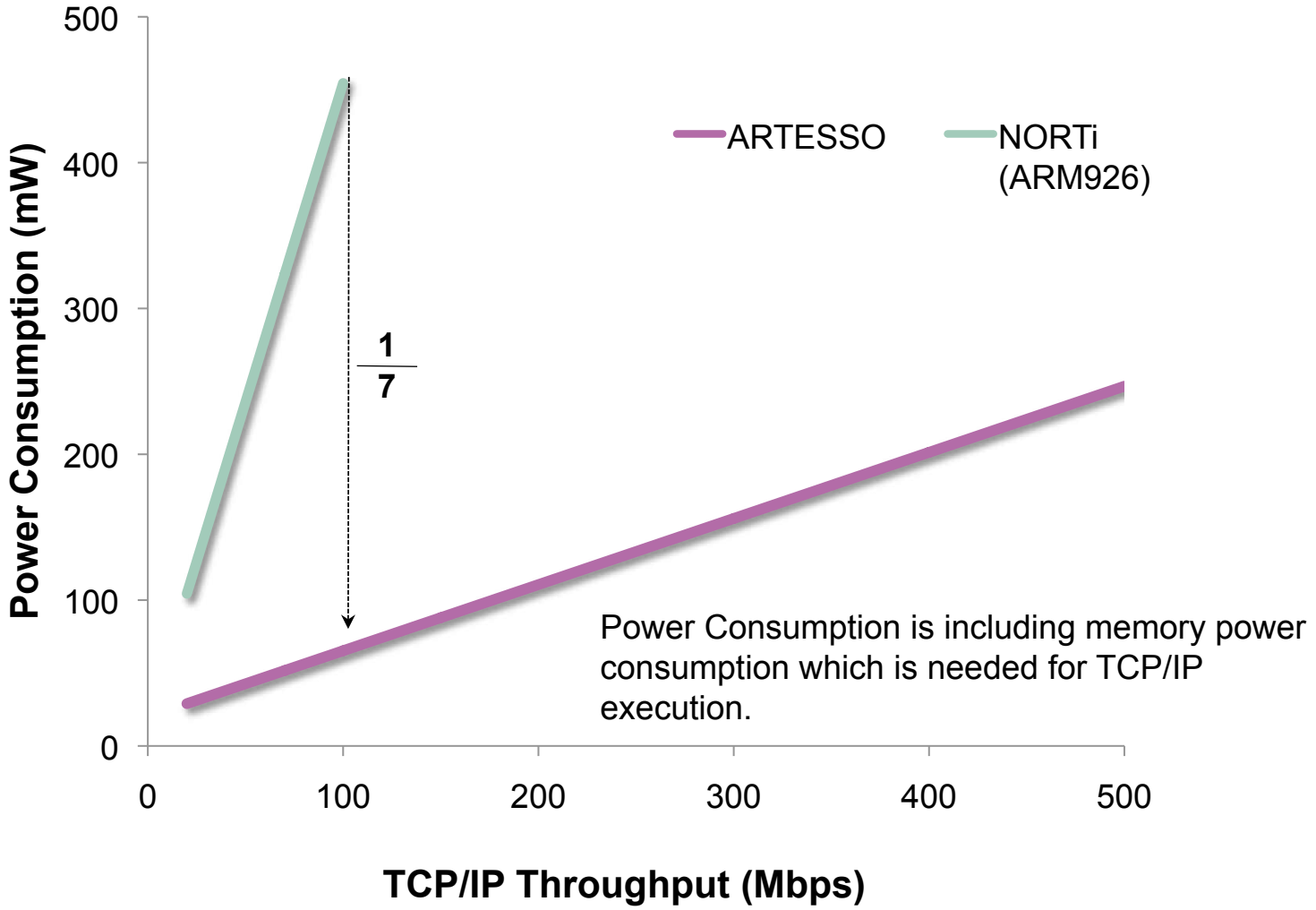
Occupation of CPU Time

	Conventional TCP/IP	ARTESSO
Protocol Processing	10.5%	92.7%
Header Rearrangement	32.1%	2.8%
Check Sum	15.3%	0%
Memory Copy	32.2%	0%
RTOS	9.9%	4.5%
	100%	100%

Achieves 125 Mbps TCP/IP throughput at 50MHz
(Conventional : 11Mbps at 50MHz)

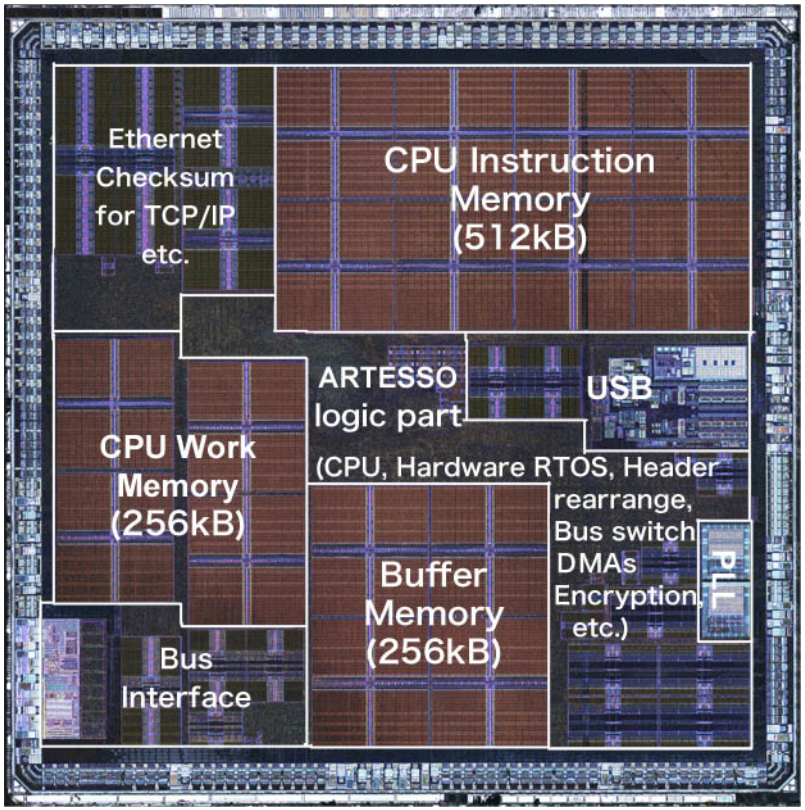
KERNELON SILICON

Throughput vs. Power Consumption

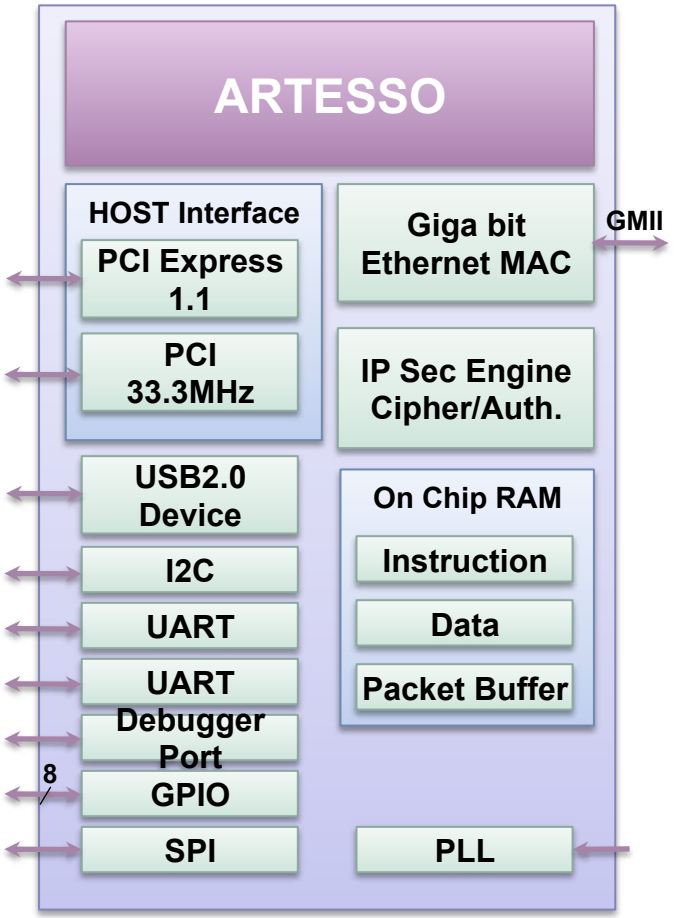


KERNELON SILICON

ARTESSO ASIC



90nmProcess ASIC



Block Diagram

It achieves a 515Mbps TCP/IP throughput at 150MHz operational clock rate.

KERNELON SILICON

Area of constituents of the ASIC

	Area (μm^2)
Total area	27,601,284
RAM for Instruction, Data, Buffer	14,851,332
ARTESSO	2,919,025
RTOS	540,170
Logic	372,964
Virtual Queue	30,213
Others	342,751
RAM	167,206
Others (CPU, Header Rearrange, DMA, Encryption, etc.)	2,378,855
Others (Ethernet, USB, Bus Interface)	9,830,927

- RTOS Spec.
  - 32 Tasks
  - 32 Semaphore IDs
  - 32 Event IDs
  - 192 Mail Box IDs
  - 16 Priorities
- Total number of wait queues are 4,096.
- Gate counts
  - RTOS 170,000
  - Virtual Queue 38,300

KERNELON SILICON

Conclusion

- **TCP/IP Processing**
  - **Several hundreds of Mbps TCP/IP needs high performance CPU.**
  - **Essential TCP/IP protocol processing needs only 10% of CPU time.**
  - **More than 30% CPU time is spending for RTOS processing.**
- **ARTESSO**
  - **RTOS and other routine processes except for protocol processing are implemented in hardware.**
    - **Essential TCP/IP process is still implement in software**
    - **Modification is available**
  - **ARTESSO achieves more than 10 times TCP/IP performance. The power consumption is 7 times lower than conventional processor.**
  - **The Virtual Queue is a key technology to realize RTOS in hardware.**