

# Introduction to TI PRUSS

## Programmable Real-time Unit Subsystem

Pierre Surply

EPITA 2016

February 11, 2014

# What is PRUSS ?

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

- Programmable Real-time Unit Subsystem
- Dual 32-bit RISC processors running at  $\frac{1}{2}$  Host CPU frequency
- Local instructions and data RAM. Access to SoC resources
- Specifically designed to implement system features that have tight real time constraints

# What is NOT PRUSS ?

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

- Not a hardware accelerator to speed up algorithm computations
- Not a general purpose RISC processor
  - No cache, no pipeline
  - No C programming
  - No OS

# PRUSS Overview

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

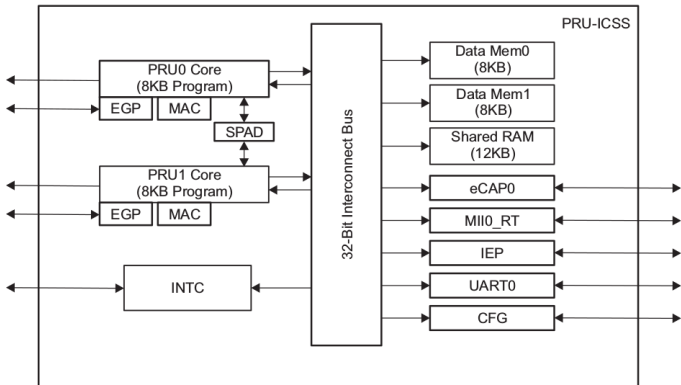


Figure : PRUSS Block Diagram

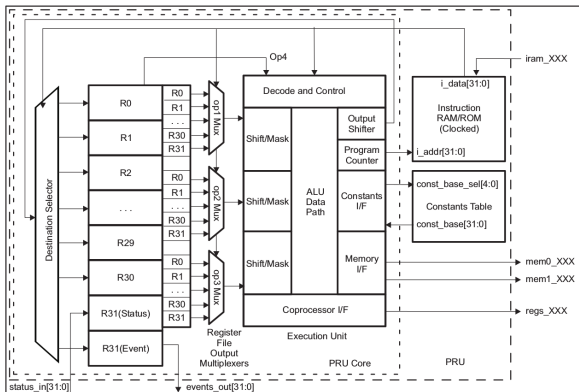


Figure : PRU Block Diagram

# PRU memory mapping

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

Start address	PRU0	PRU1
0x00000000	Data RAM0 8KB	Data RAM1 8KB
0x00002000	Data RAM1 8KB	Data RAM0 8KB
0x00010000	Shared RAM 12KB	Shared RAM 12KB

## Arithmetic and Logical

ADD	ADC	SUB	SUC	RSB	RSC
LSL	LSR	AND	OR	XOR	NOT
MIN	MAX	CLR	SET	LMBD	NOP

## Load/Store

MOV	LDI	MVI <sub>x</sub>	LBBO	SBBO	LBCO
SBCO	ZERO	FILL	XIN	XOUT	XCHG
SXIN	SXOUT	SXCHG			

## Flow Control

JMP	JAL	CALL	RET	QBGT	QBGE
QBLT	QBLE	QBEQ	QBNE	QBA	QBBS
QBBC	WBS	WBC	HALT	SLP	LOOP

```
$ lsmod | grep pruss
uio_pruss                3825  0
$ ls /dev/uio*
/dev/uio0  /dev/uio1  /dev/uio2  /dev/uio3
/dev/uio4  /dev/uio5  /dev/uio6  /dev/uio7
```



# Using GPIO via SysFS

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

```
$ echo 46 > /sys/class/gpio/export
$ ls /sys/class/gpio/
export      gpio46      gpiochip0
gpiochip32  gpiochip64  gpiochip96
unexport
$ ls /sys/class/gpio/gpio46
active_low  direction  edge  power
subsystem  uevent     value
$ cat /sys/class/gpio/gpio46/direction
in
$ cat /sys/class/gpio/gpio46/value
0
$ cat /sys/class/gpio/gpio46/value
1
```

# Using GPIO via PRU

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

```
#define GPIO_PIN    15
.origin 0
.entrypoint start

start:  XOR    R0, R0, R0
        XOR    R1, R1, R1
        SBBO   R1, R0, 0, 4
l1:     WBS    R31, GPIO_PIN
        ADD    R1, R1, 1
        SBBO   R1, R0, 0, 4
        WBC    R31, GPIO_PIN
        JMP    l1
```

# GPIO Benchmark (Freq = 2 Hz)

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

- Sysfs: 10
- Mmap: 10
- PRU: 10
  - $\frac{10}{5} = 2Hz$

# GPIO Benchmark (Freq = 1.6 MHz)

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

- Sysfs: 532922
- Mmap: 7754509
- PRU: 7991682
  - $\frac{7991682}{5} \simeq 1598336.4 \simeq 1.6MHz$

Introduction  
to TI PRUSS

Pierre Surply

Introduction

PRUSS

PRU

Linux driver

Application

Conclusion

- Interrupt Controller
- UART
- Industrial Ethernet Peripheral