DEVELOPMENT AND PERFORMANCE ANALYSIS OF EPICS CHANNEL ACCESS SERVER ON FPGA BASED SOFT-CORE PROCESSOR

PCaPAC 2012

Shantonu Sahoo

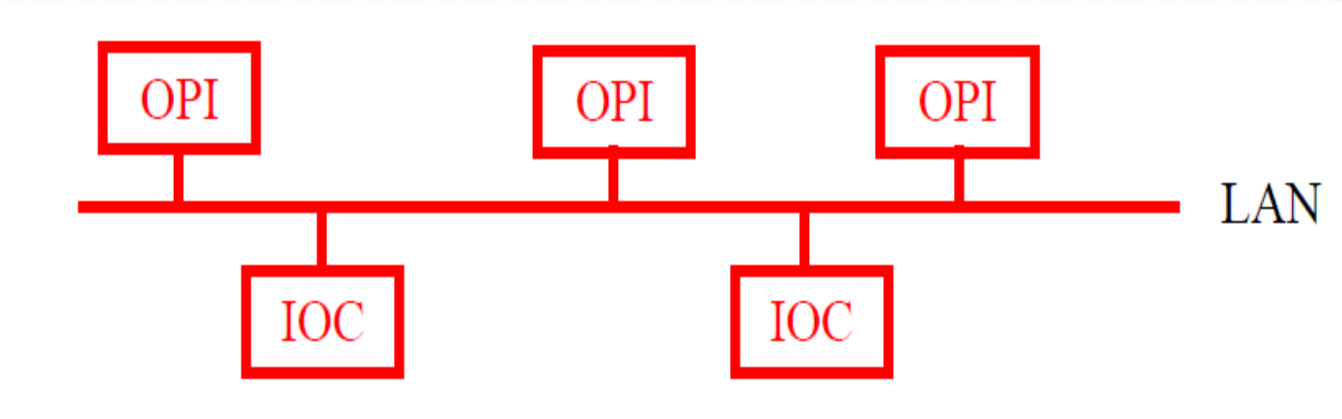
Computer & Informatics Group

Variable Energy Cyclotron Centre

Discussion Outline:

- Embedded EPICS system
- Selection of Soft-core processor.
- Porting of EPICS on the target processor.
- EPICS performance analysis.
- Proposed Improvements.

Overview of EPICS Architecture



- **OPI (Operator Interface)** A UNIX- or NT-based workstation or PC which can run various EPICS tools—the "clients."
- **IOC (Input Output Controller)** A server containing a processor with various I/O modules for analog and digital signals.
- **LAN (TCP/IP-based Local Area Network)** A communication network which connects the IOCs and OPIs.

Advantages of porting EPICS on Embedded Systems:

1. Cost
2. Size
3. Flexibility of designing EPICS embedded control hardware or instrumentation
4. Easy Plug-in for customization
5. Real Time Performance

FPGA based soft-core processor

What is a Soft-core Processor ?

A hardware description language (HDL) model of a specific processor (CPU) that can be customized for a given application and synthesized for an ASIC/FPGA target.

Technology	Performance/Cost	Time until running	Time to high performance	Time to change code functionality
ASIC	Very High	Very Long	Very Long	Impossible
Custom Processor/ DSP	Medium	Long	Long	Long
FPGA	Low-Medium	Short	Short	Short
Generic	Low-Medium	Short	Not Attainable	Short

Speed

Flexibility

Speed-flexibility Trade-off in Embedded Systems

Which soft-core processor to select?

CPU core	Architecture	Bits	License	Pipeline depth	MMU	MUL	FPU	Area (LEs)	Comments
S1 Core	SPARC-v9	64	Open-source (GPL)	6	+	+	+	37000 - 60000	Single-core version of UltraSPARC T1
LEON3	SPARC-v8	32	Open-source (GPL)	7	+	+	+	3500	
OpenRISC 1200	OpenRISC 1000	32	Open-source (LGPL)	5	+	+	-	6000	
MicroBlaze	MicroBlaze	32	Proprietary	3, 5	opt	opt	opt	1324	Limited to Xilinx devices
aeMB	MicroBlaze	32	Open-source (LGPL)	3	-	opt	-	2536	Open-source clones of MicroBlaze
OpenFire	MicroBlaze	32	Open-source (MIT)	3	-	opt	-	1928	
Nios II/f	Nios II	32	Proprietary	6	+	+	opt	1800	Limited to Altera devices
Nios II/s	Nios II	32	Proprietary	5	-	+	opt	1170	
LatticeMico32	LatticeMico32	32	Open-source	6	-	opt	-	1984	Not limited to Lattice devices (can be used elsewhere)
Cortex-M1	ARMv6	32	Proprietary	3	-	+	-	2600	
PicoBlaze	PicoBlaze	8	Proprietary, zero-cost	no	-	-	-	192	Limited to Xilinx devices
LatticeMico8	LatticeMico8	8	Open-source	no	-	-	-	200	Not limited to Lattice devices (can be used elsewhere)

Advantages of soft-core processors:

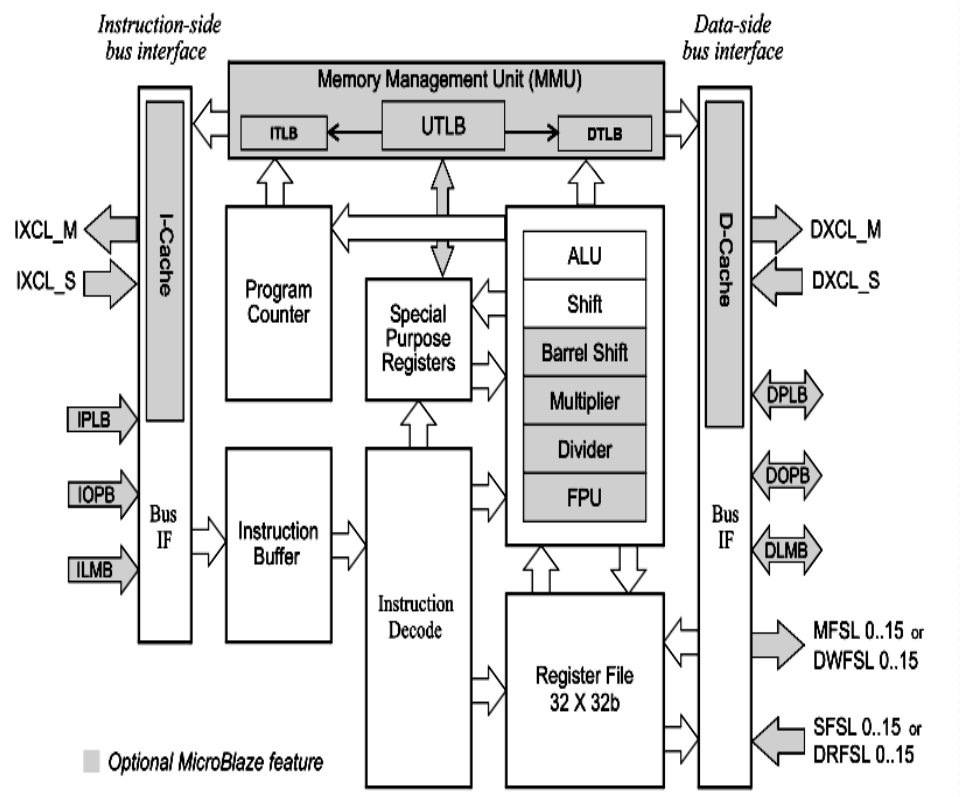
- Flexibility / Reconfigurability
- Peripherals integration
- Obsolescence mitigation
- Component and cost reduction
- Hardware acceleration
- Education and research on Microprocessor internal architecture.

Disadvantages of soft-core processors:

- Performance
- Power Consumption
- Increased Design Complexity
- Compiler/linker/debugger toolchain

MicroBlaze soft-core processor

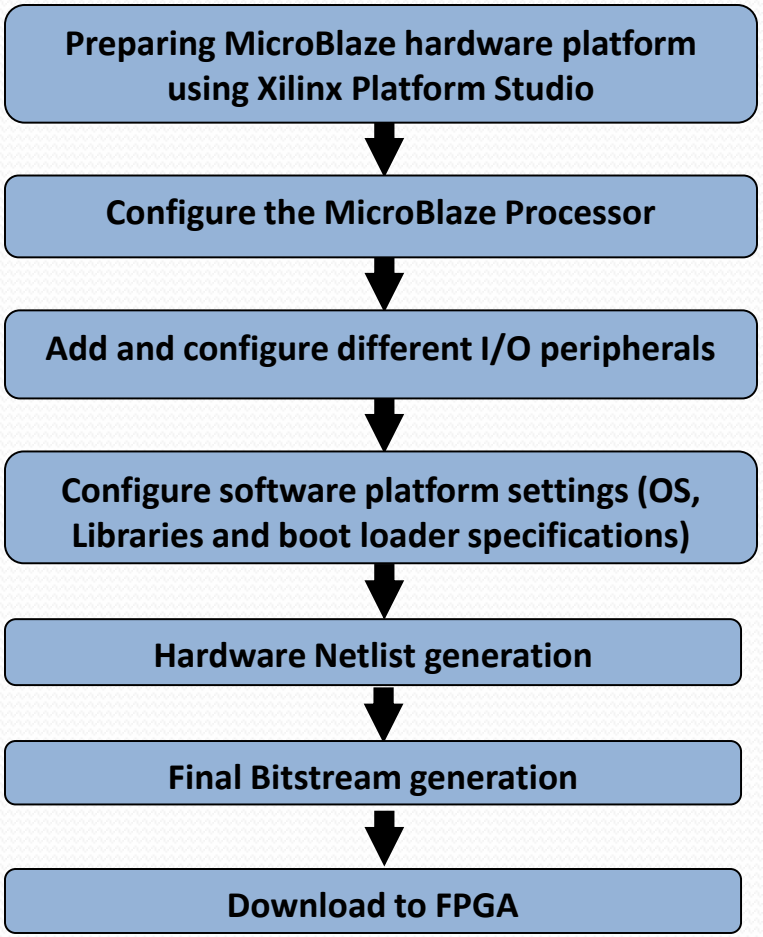
- Thirty-two 32-bit General Purpose Registers (R0 to R31)
- 32 Bit RISC Instruction Set
- Big-Endian bit-reversed Data Format
- Harvard Memory Architecture
- Pipelined Architecture
- Optional Memory Management Unit
- Optional Instruction and Data Cache
- Optional Floating Point Unit



Steps required:

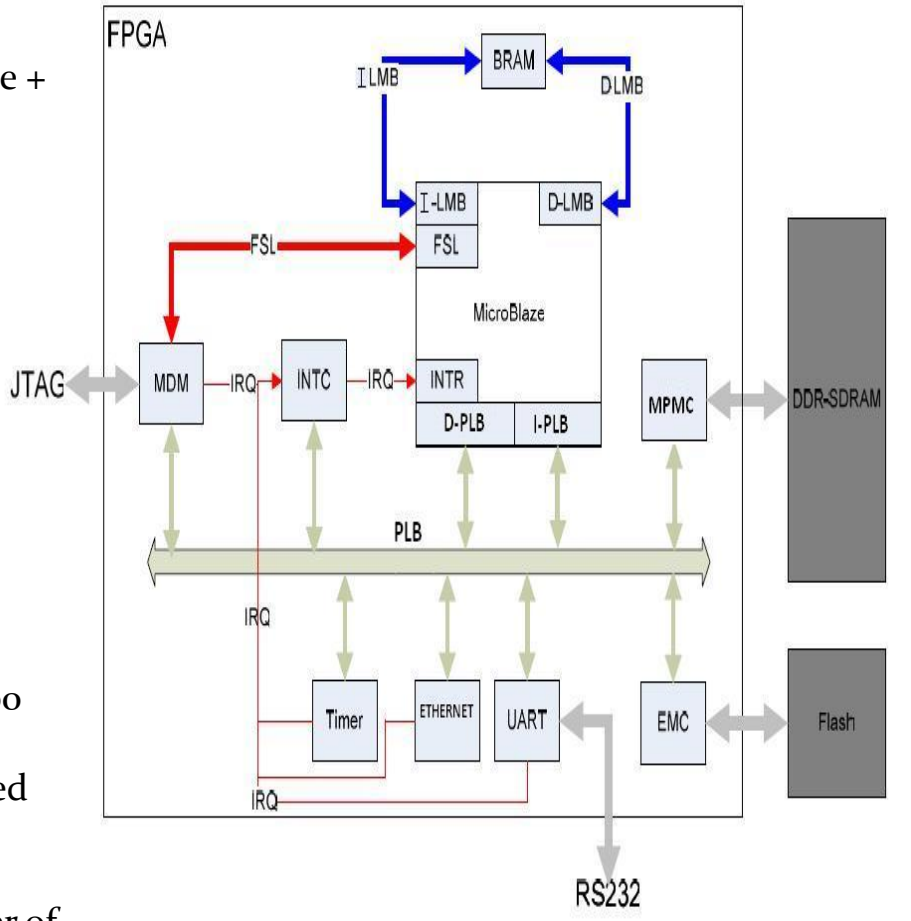
1. Building Soft-core Processor on FPGA
2. Porting Operating System
3. Porting EPICS

Building MicroBlaze Soft-core Processor on Xilinx FPGA



MicroBlaze Processor:

Processor Clock Frequency : 62.50 Mhz
Local Memory (Block RAM) : 8 KB
Cache Memory (Block RAM): 4 KB (2KB Data Cache + 2KB Instruction Cache)
Total Off Chip Memory : 144 MB
DDR2 SDRAM = 128 MB
FLASH = 16 MB
No Memory Management Unit
3 Stage Pipelining (Size Optimized)
No Floating Point Unit
32 Bit Barrel Shifter
32 Bit Integer Multiplier



Peripherals:

1. RS232 UARTLITE: Baudrate (bits per second) : 9600
2. ETHERNETLITE: Link Speed : 10/100 Mbps
3. MPMC (Multi-Port Memory Controller): Controlled interface for external (off-chip)FLASH memory
4. TIMER: Counter Bit Width : 32 Bits
5. INTC (Interrupt Controller) Configurable number of (up to 32) interrupt inputs and a single interrupt output

Base System Builder - System Created

Below is a summary of the system you have created. Please review the information below. If it is correct, hit <Generate> to enter the information into the XPS data base and generate the system files. Otherwise return to the previous page to make corrections.

Processor: microblaze_0
System clock frequency: 62.50 MHz
On Chip Memory : 8 KB
Total Off Chip Memory : 144 MB
- DDR2_SDRAM = 128 MB
- FLASH = 16 MB

The address maps below have been automatically assigned. You can modify them using the editing features of XPS.

PLB Bus : PLB_V46 Inst. name: mb_plb Attached Components:

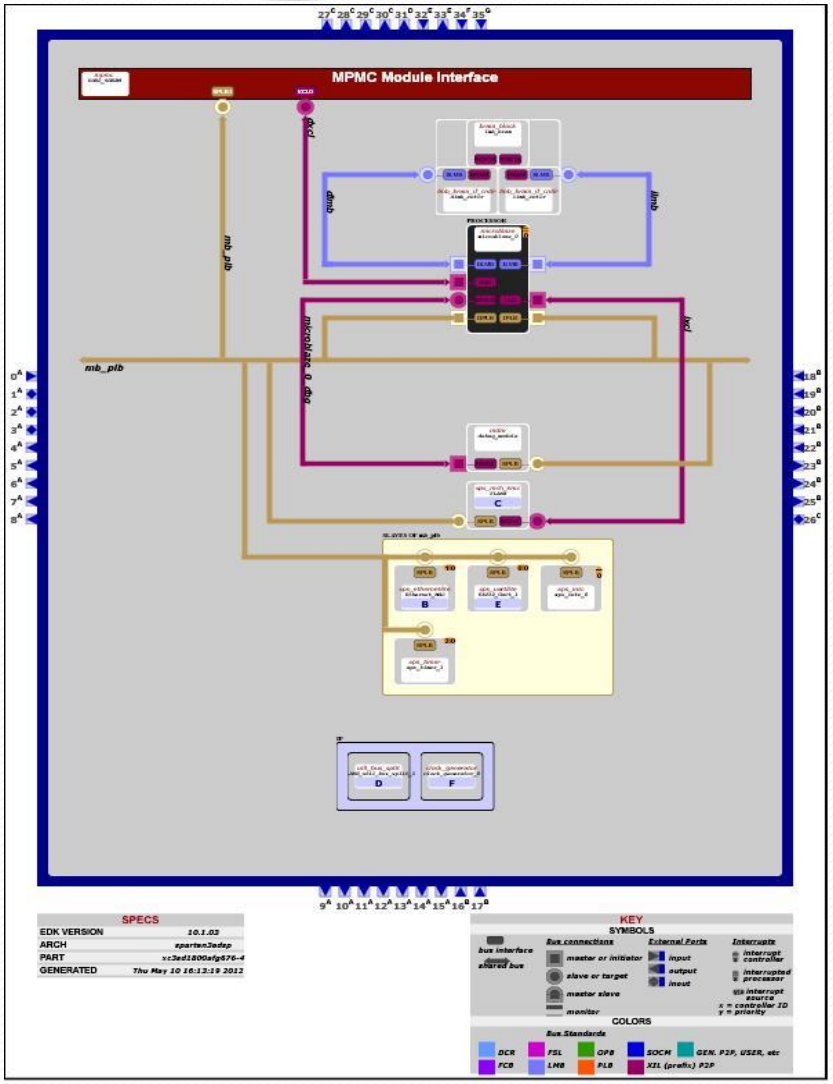
Core Name	Instance Name	Base Addr	High Addr
xps_uartlite	RS232_Uart_1	0x84000000	0x8400FFFF
xps_ethernetlite	Ethernet_MAC	0x81000000	0x8100FFFF
mpmc	DDR2_SDRAM_C_MP	0x88000000	0x8FFFFFFF
xps_mch_emc	FLASH	0x87000000	0x87FFFFFF
xps_timer	xps_timer_1	0x83C00000	0x83C0FFFF
mdm	debug_module	0x84400000	0x8440FFFF
xps_intc	xps_intc_0	0x81800000	0x8180FFFF

LMB Bus : LMB_V10 Inst. name: ilmb Attached Components:

Core Name	Instance Name	Base Addr	High Addr
lmb_bram_if_cntlr	ilmb_cntlr	0x00000000	0x00001FFF

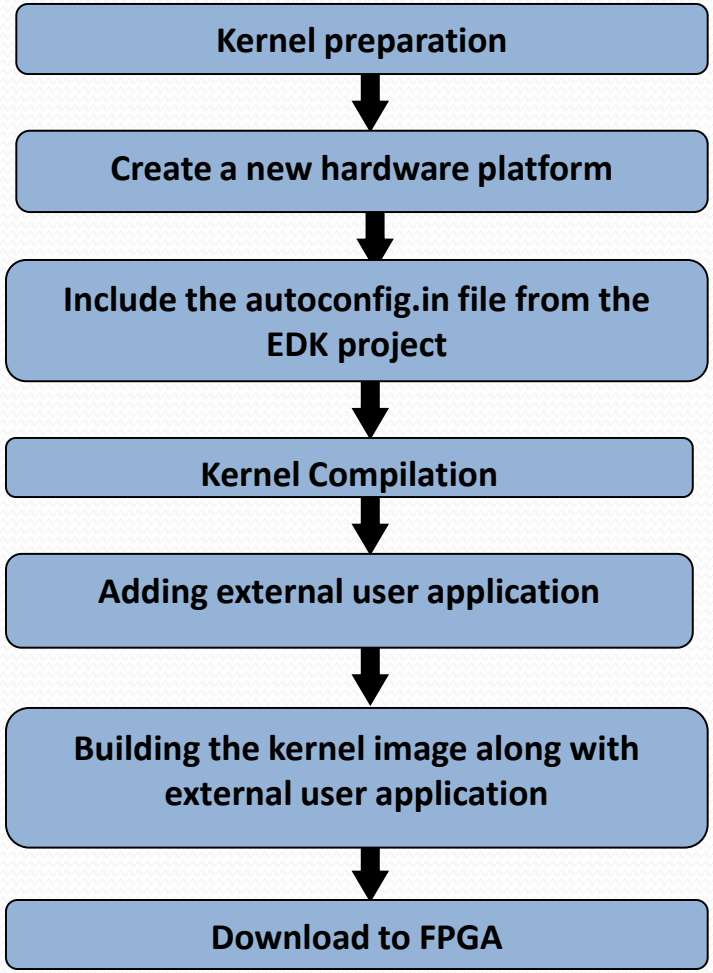
LMB Bus : LMB_V10 Inst. name: dlmb Attached Components:

Core Name	Instance Name	Base Addr	High Addr
lmb_bram_if_cntlr	dlmb_cntlr	0x00000000	0x00001FFF

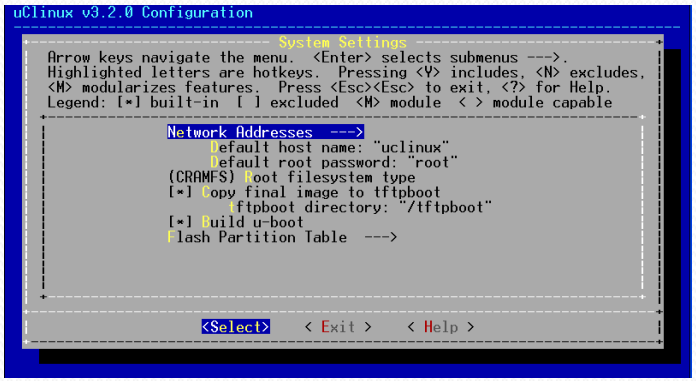


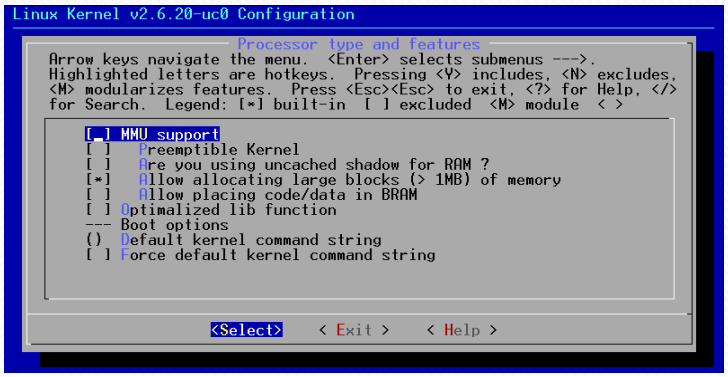
Screenshot of the Final MicroBlaze system designed in Xilinx Platform Studio

Porting uClinux Kernel on MicroBlaze

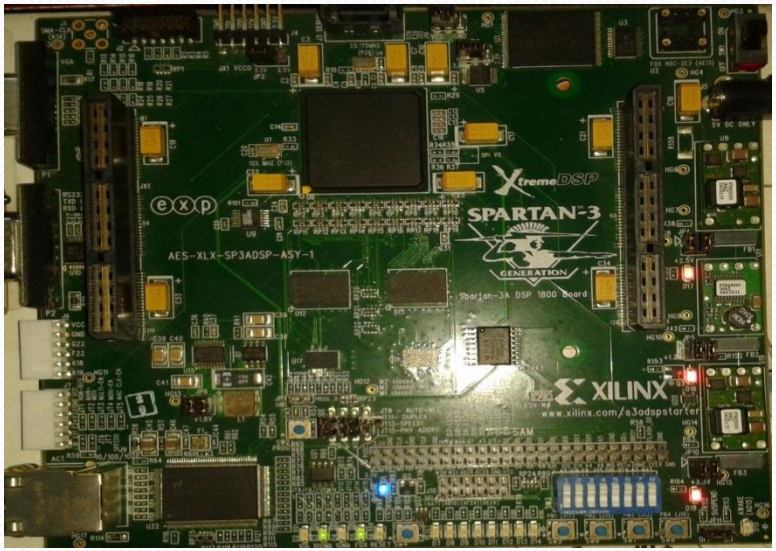


Some screenshots:

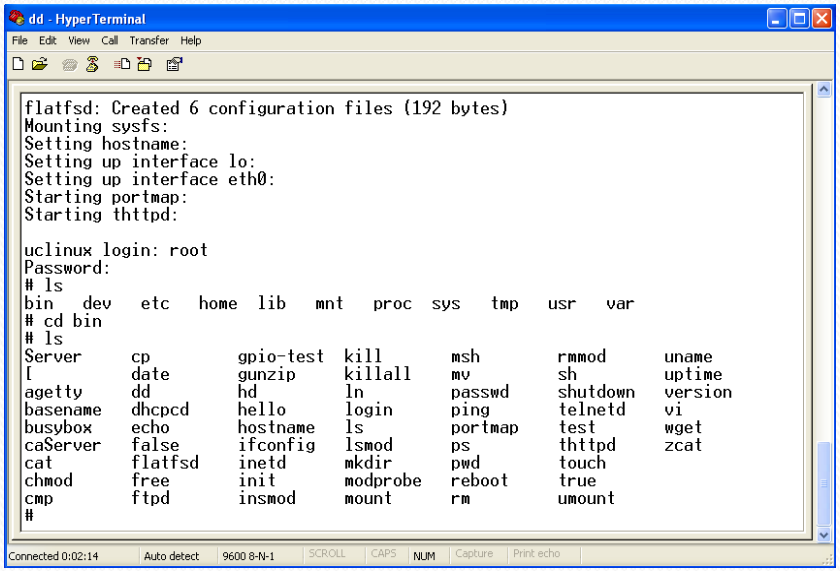




Kernel Compilation of uCLinux for MicroBlaze



Xilinx Spartan 3A DSP 1800 board



Hyperteminal screenshot of console terminal

Porting of EPICS on MicroBlaze

1. Customizing C/C++ codes for Portable Channel Access Server
2. Building GNU Cross Compiler toolchain for MicroBlaze-uClinux platform (binutils-2.16, gcc-4.1.2, gbd-6.5, newlib-1.14.0)
3. Building the necessary Library Packages for MicroBlaze-uClinux platform (libCom, libca, libcas, libgdd, librt)
4. Compiling the C/C++ codes using MicroBlaze-uClinux toolchain
5. Building the uClinux kernel image along with server application
6. Downloading the kernel image into FPGA

EPICS Performance Analysis

In the performance analysis of Channel Access Server we have calculated mainly two parameters:

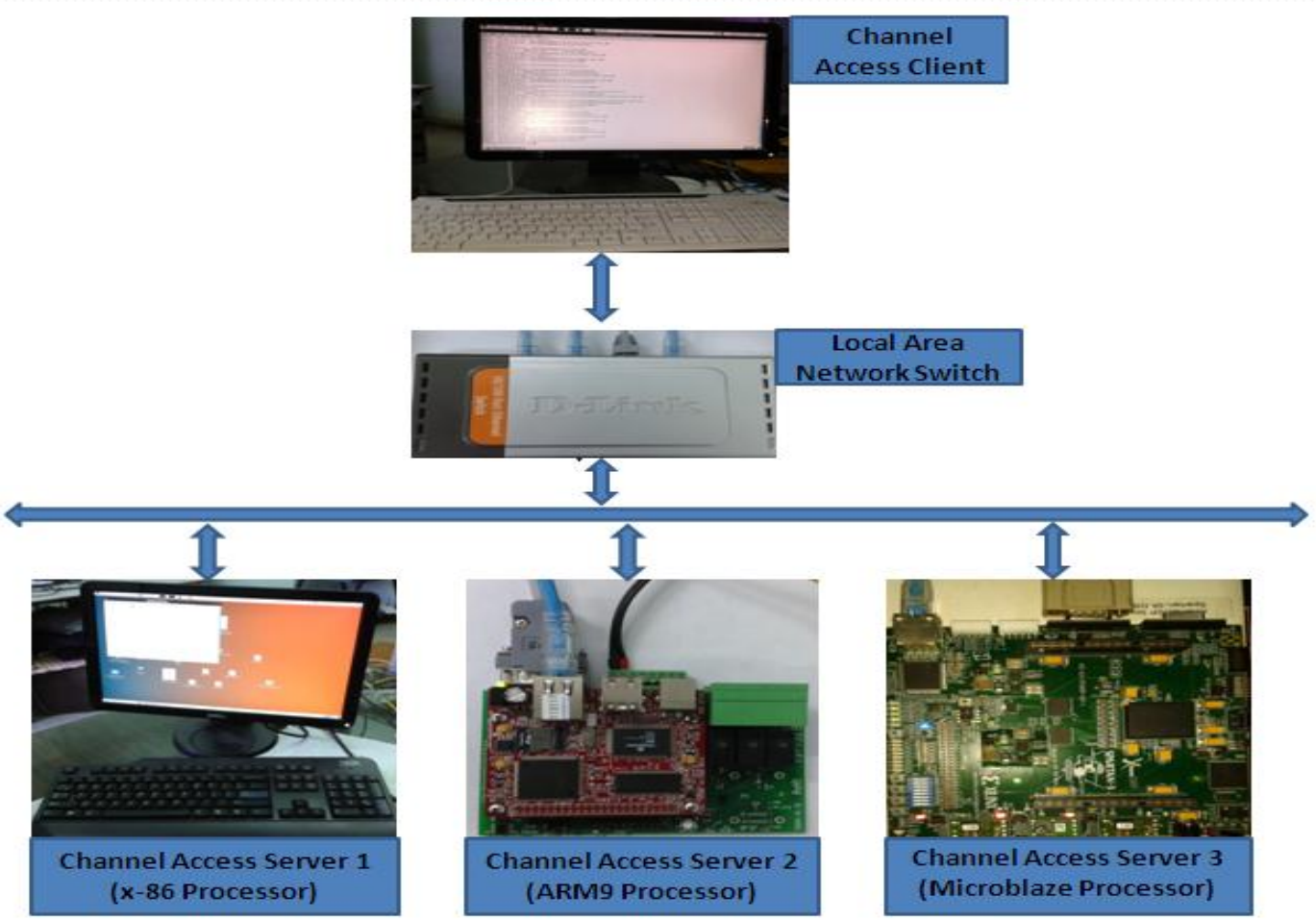
- **Server CPU Load** : Percentage utilization of CPU resource at the server end while servicing to the client requests.
- **Server Processing Time**: Time required by the server to retrieve, process and serve the client requests.

Platforms in which EPICS Channel Access Performance is Tested:

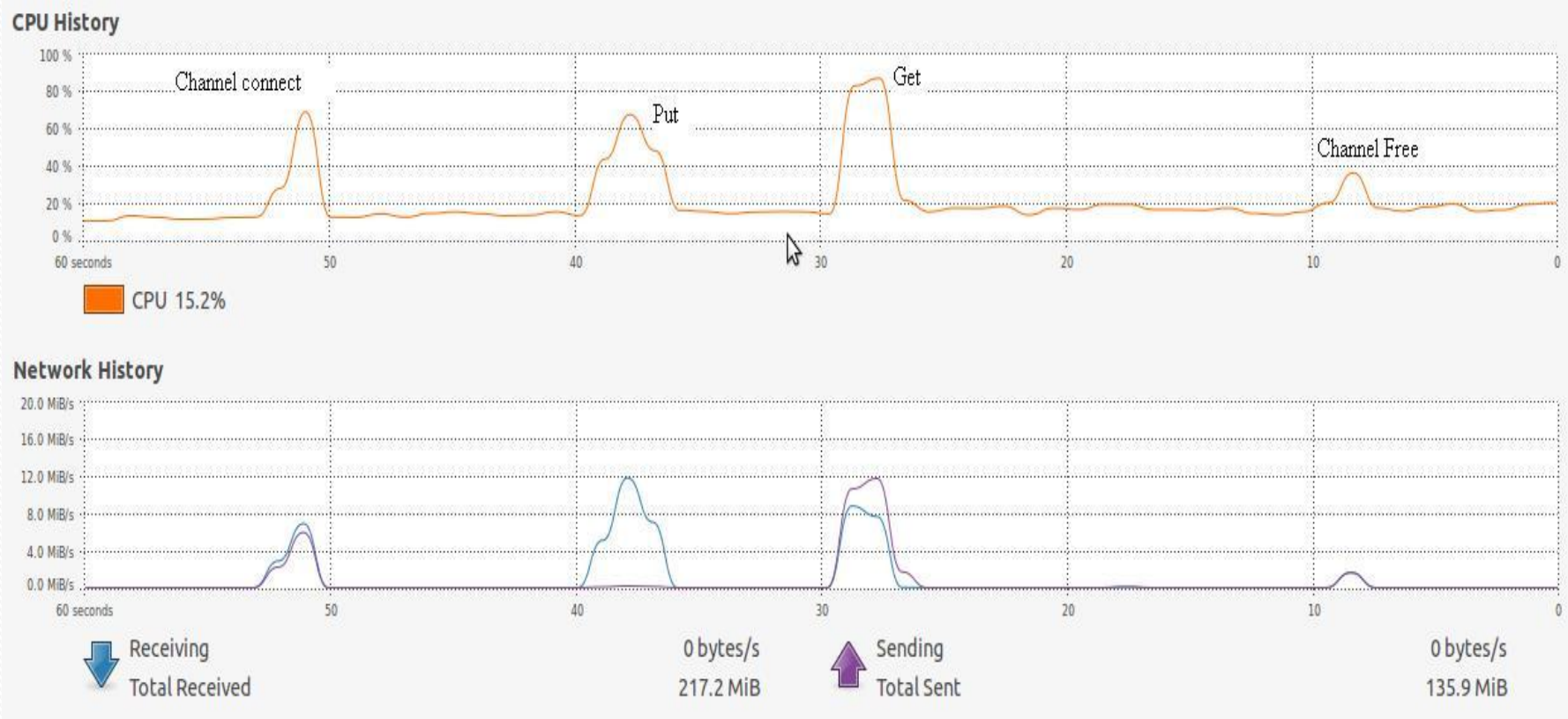
1. **x-86 processor**
2. **ARM processor**
3. **MicroBlaze Processor**

	Pipelining	I-Cache	D-Cache
Configuration -1	3 Stage	2 KB	2 KB
Configuration -2	3 Stage	8 KB	8 KB
Configuration -3	5 Stage	2 KB	2 KB
Configuration -4	5 Stage	8 KB	8KB

Experimental setup for EPICS Performance Analysis

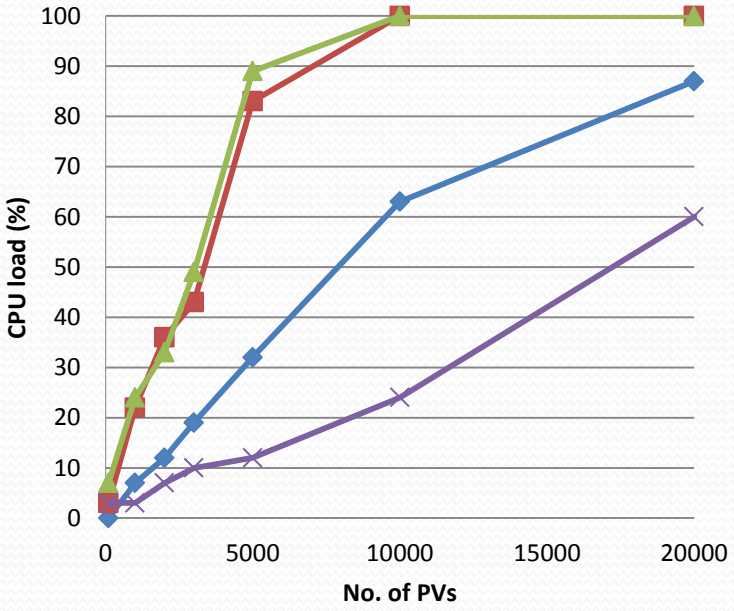


CPU and Network report of x-86 Server machine demonstrating the performance of an active CAS

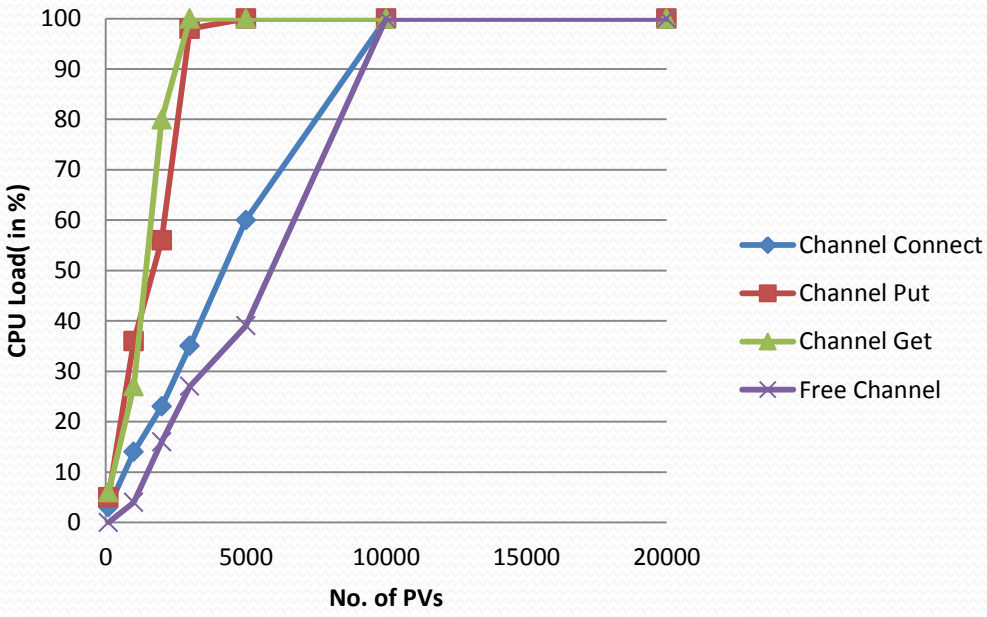


* No. of PVs accessed simultaneously : 100000

CPU load performance analysis

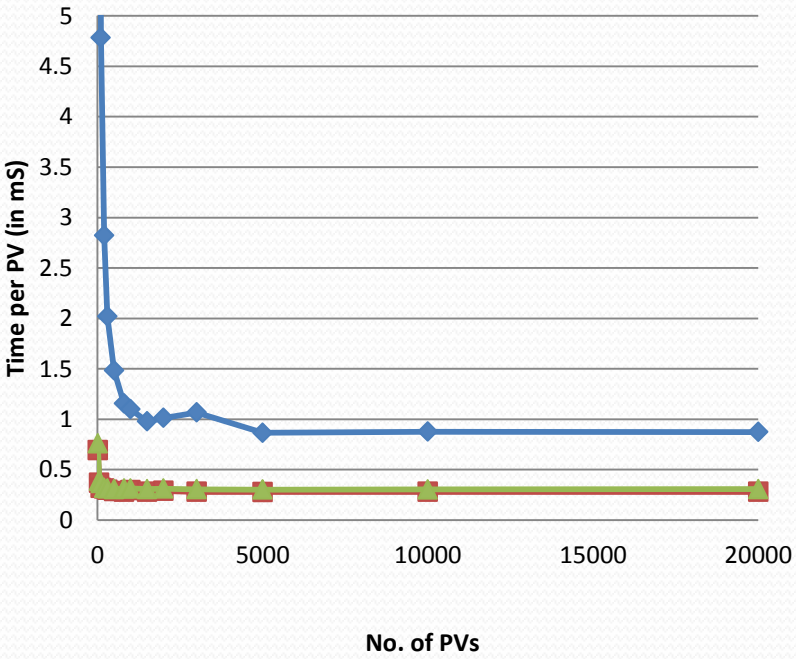


ARM9 Processor

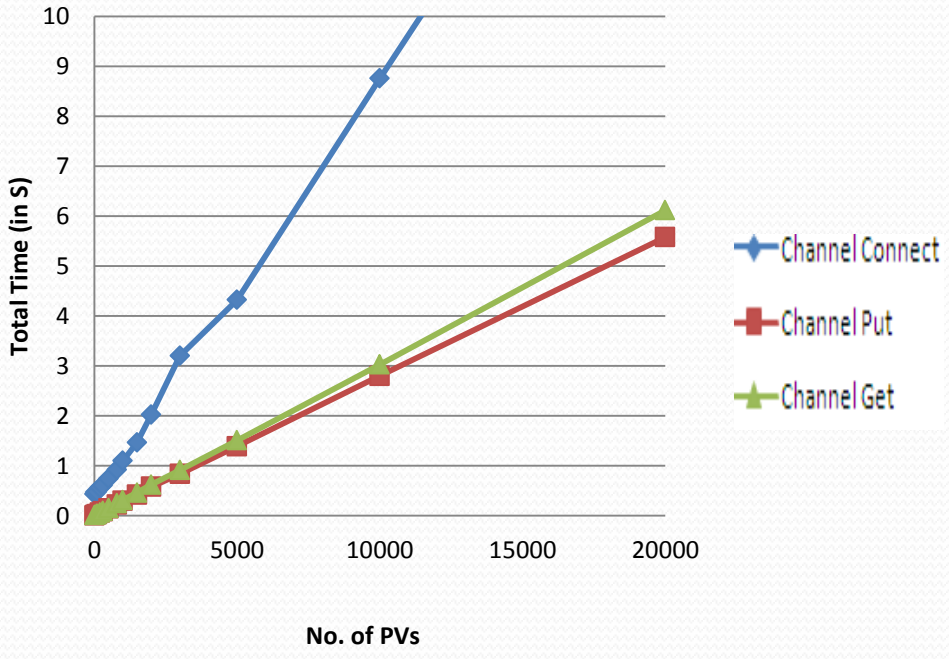


MicroBlaze Processor (Conf. 1)

Server Processing Time- ARM

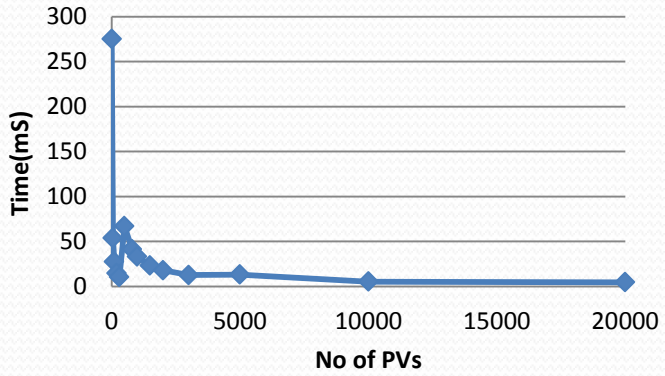


Server Processing Time per PV

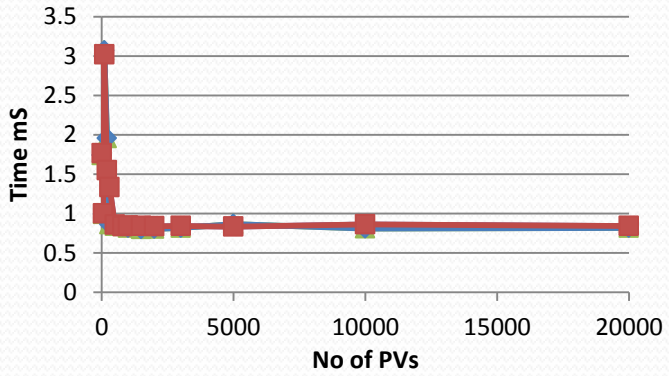


Total Server Processing Time

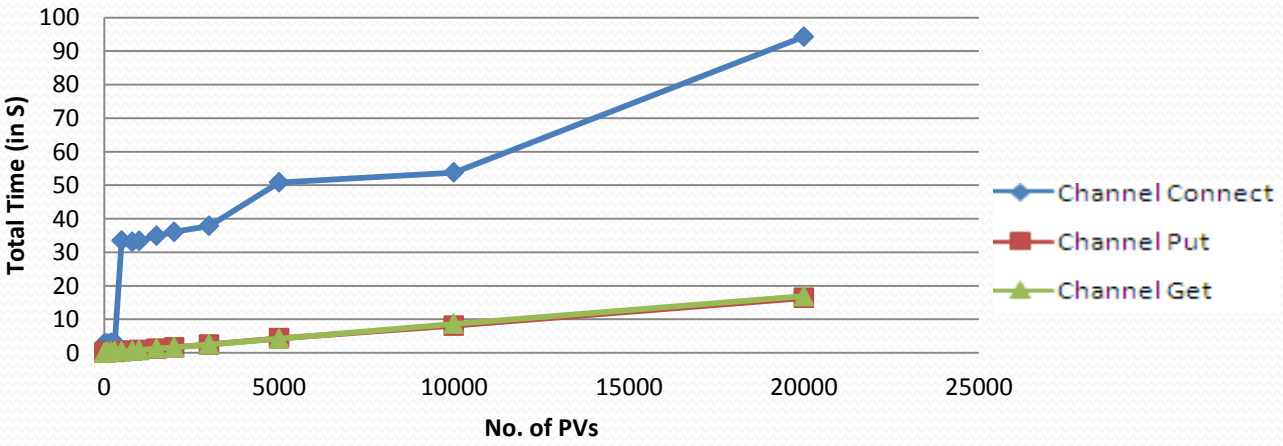
Server Processing Time- MicroBlaze (Conf. 1)



(a): Processing Time per PV in Channel Connect

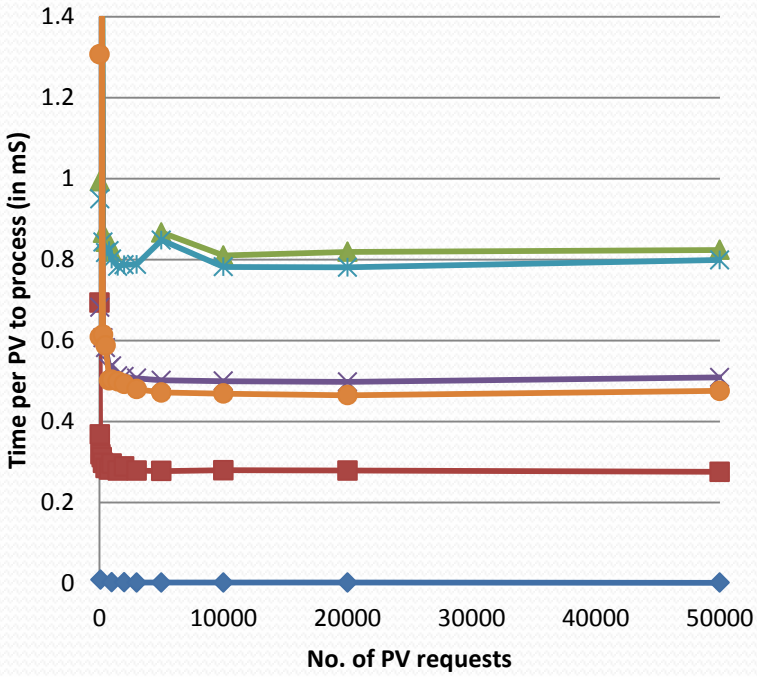


(b) Processing Time per PV in Channel Put and Get

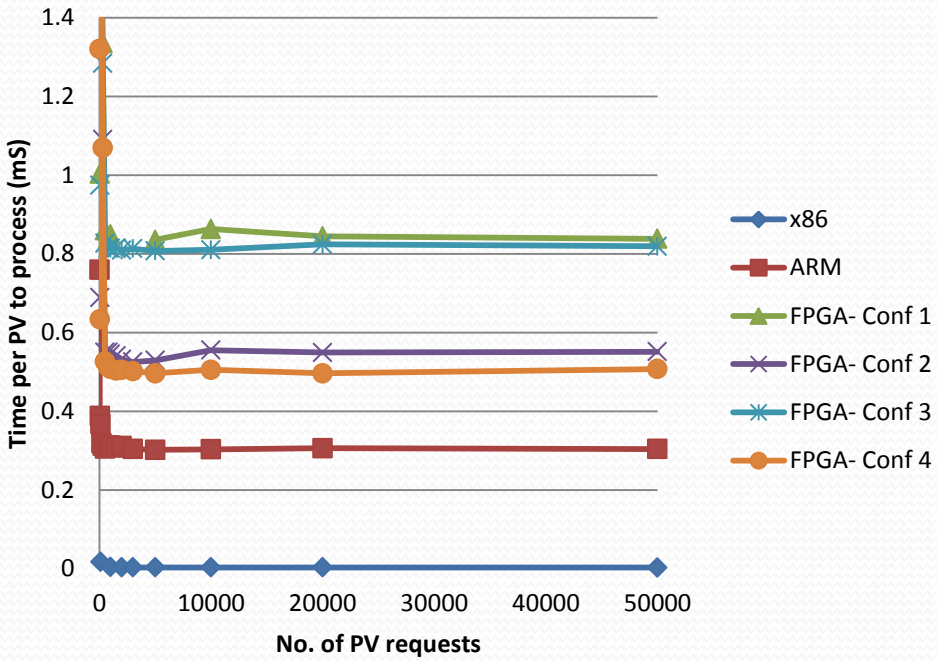


(c): Total Processing Time

Performance Comparison on different architectures



Channel Put

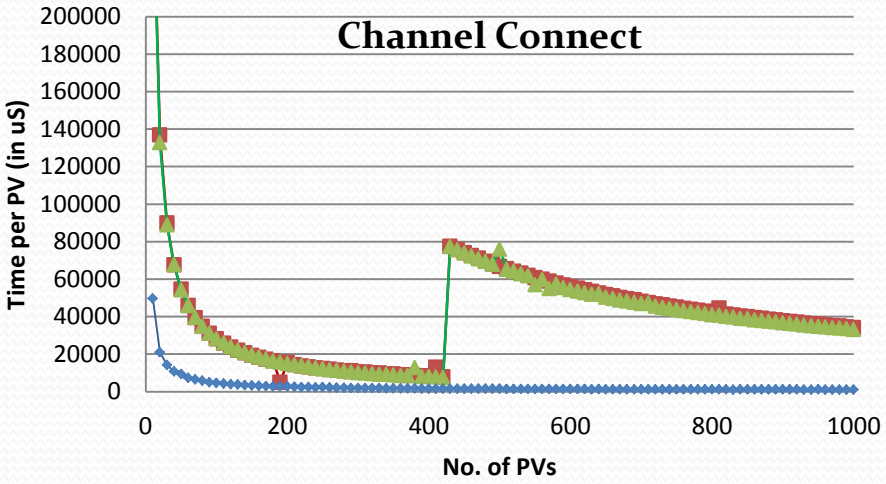


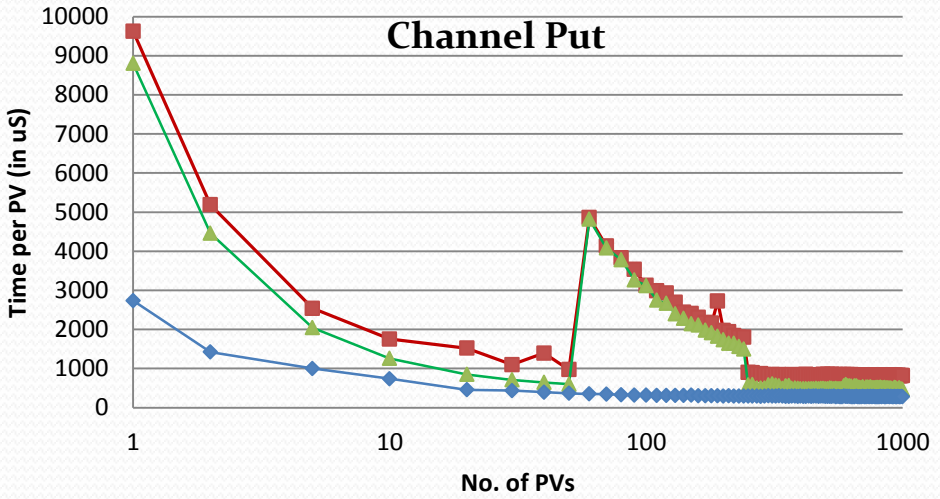
Channel Get

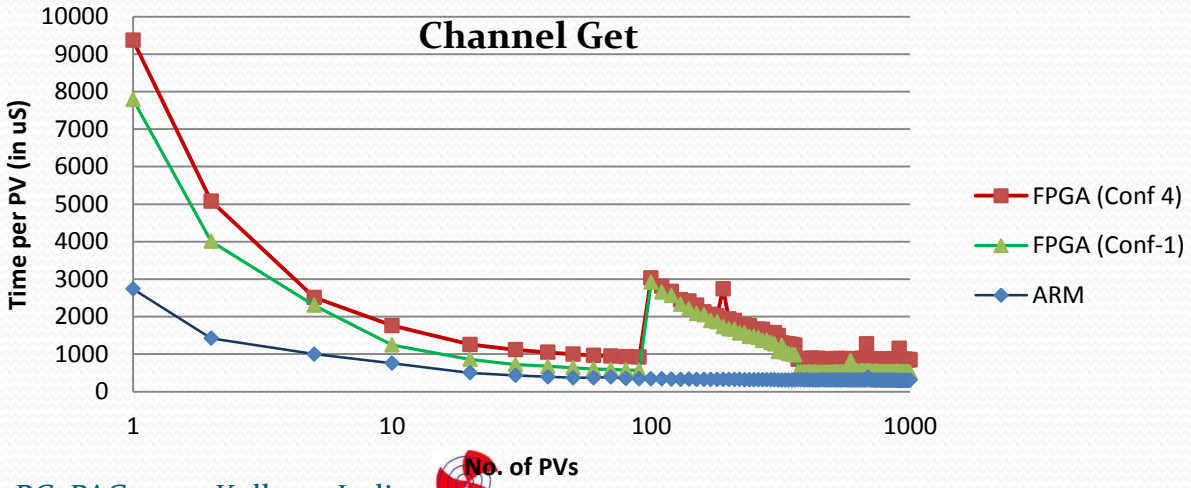
Results*:

	Max. PV Limit (SCAN Period=0.1 s)	Max PV Limit (SCAN Period=1 s)	Max PV Limit (SCAN Period=10s)	Safe PV Limit (Without overloading server CPU)
X86	50,000	5,00,000	50,00,000	2,00,000
ARM9	400	4,000	40,000	5,000
Microblaze (2kB Cache +3 stage PP)	80	1,200	12,000	2,500
Microblaze (8kB Cache +3stage PP)	80	2,000	20,000	3,000
Microblaze (2kB Cache +5 stage PP)	80	1,300	13,000	2,500
Microblaze (8kB Cache +5stage PP)	80	2,200	22,000	4,000

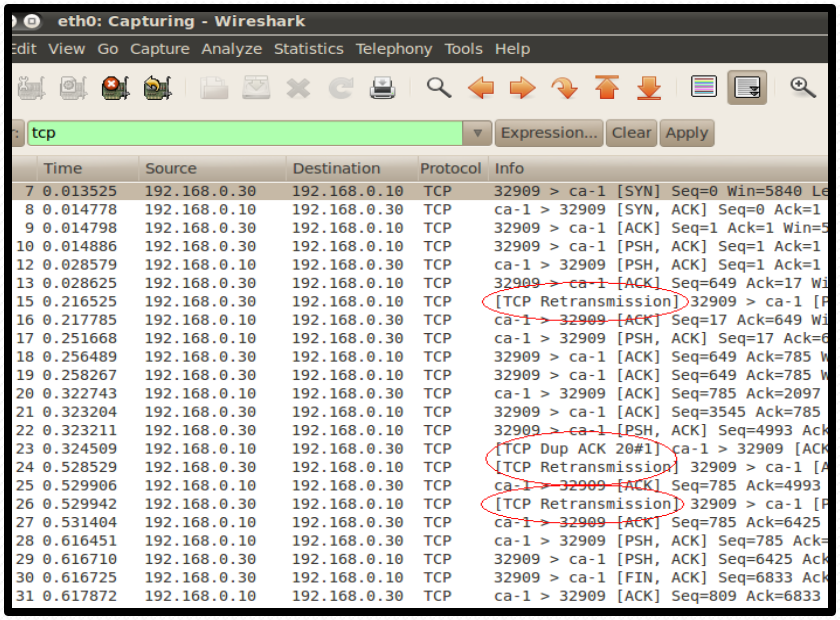
Abnormal Behavior of MicroBlaze processor in Lower range of PVs:



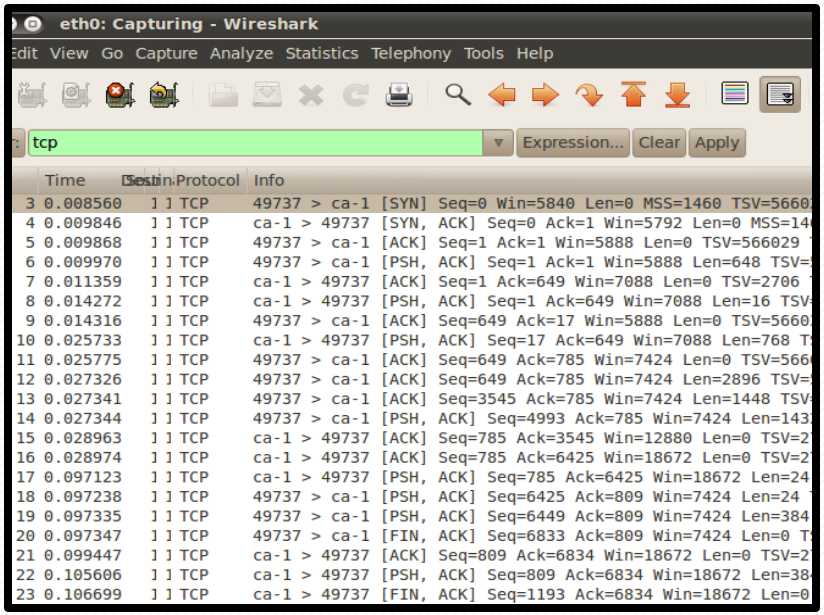




Wireshark screenshot showing the TCP communication



MicroBlaze Processor



ARM Processor

No of PVs: 240

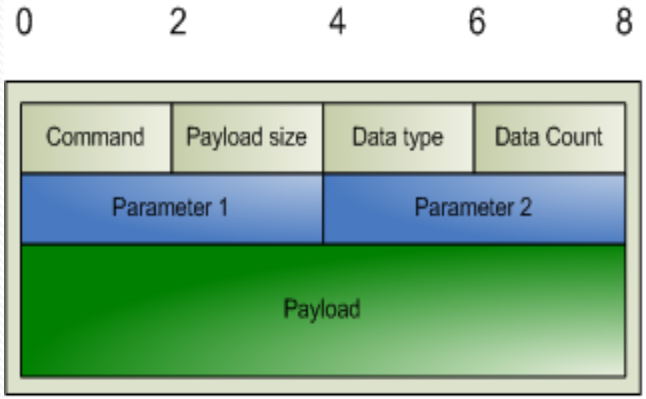
Proposed Improvements

- Start with a fixed payload size say Payload $_{Original}$ for channel access message buffer.
- If retransmissions are detected in the media, reduce the payload size to half of its previous value.

  i.e. Payload $_{New}$ = Payload $_{Original}$ / 2
- Detect for any retransmission in the media. If retransmission is detected reduce the payload size further by half of its previous value.
- Thus after N iterations when there is no retransmission in the media, the final payload size becomes:

  Payload $_{Final}$ = Payload $_{Original}$ / 2^N

$$N = \log_2 \frac{\text{Payload}_{\text{Original}}}{\text{Payload}_{\text{Final}}}$$



General Message Buffer Structure in Channel Access

Conclusion

- In this project we have successfully ported EPICS channel access server on MicroBlaze soft-core processor.
- EPICS channel access and record processing performances have been analyzed for MicroBlaze platform and the results are compared with standard ARM9 processor.
- The performance in MicroBlaze soft-core processor can be considerably improved by suitably tuning the processor architecture (like size of cache memory and number of stages of pipelining).
- An improvement in Channel Access Protocol has been proposed for embedded system. The size of message buffer in EPICS channel access can be optimized to reduce the rate of re-transmissions in communication line.

Future scope:

- A few more performance metrics for soft real-time performance metrics are required to be derived:
    1. Interrupt latency
    2. EPICS event latency.

- The interrupt latency test of the EPICS IOC-Core provide the information of the total processing time of an epics record when interrupted by an external hardware interrupt.
    1. Interrupt Top Half to Bottom Half latency
    2. Interrupt Bottom Half to EPICS record processing latency
    3. Context switch latency in multi-tasking environment

- The event latency measures the processing time when the record generates an interrupt for the external hardware device.

Acknowledgement

- **Shri Tanushyam Bhattacharjee**
- **Dr. Sarbajit Pal**

Thank You