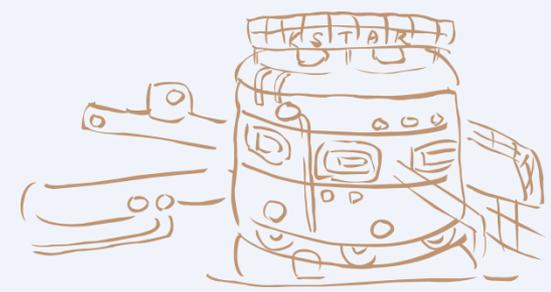


EPICS collaboration meeting fall 2012  
22-26 October 2012  
by Pohang Accelerator Laboratory (PAL)



# Application with RT-patched EPICS for real-time monitoring

*Fusion  
Future Vision of Green Energy*

**October 23, 2012**

**Woongryol Lee**



**NFRI** 국가핵융합연구소  
National Fusion Research Institute

## I. Introduction

A. Motivation

B. Background

## II. Implementation

A. System configuration

B. Operation flow and optimization

## III. Test results

## IV. conclusion and future work

## ● Motivation

### ◆ Supervisory monitoring system to catch up the RT data

- ✓ Reflective memory card for the feedback control
- ✓ Analysis and new design about RT-network for the next fast control system
- ✓ Inspection tool for our feedback control interface

### ◆ Fast feedback control system in KSTAR

- ✓ Two kinds of RTOS in KSTAR
  - VxWorks : Local control system of magnetic power supply
  - Customized Linux : Plasma control system
- ✓ Improvement for the long pulse operation
  - All outbound interface of PCS was disconnected at RT-mode
  - Give a reconfigurable mechanism to PCS during plasma discharging time

### ◆ ITER standard RTOS

- ✓ We are in progress of CODAC Technologies evaluation project in KSTAR

## ● Background

### ◆ Approaches to increase real time performance

- ✓ Application binding
  - restricting certain CPUs to running designated application processes
- ✓ Interrupt binding
  - designating specific CPUs handle device interrupts
- ✓ Memory pinning
  - designating that physical memory be exclusively allocated to dedicated processes.
- ✓ Scheduler priority control
  - ability to designate process priorities at a fine grained level

### ◆ Red Hat™ real-time OS

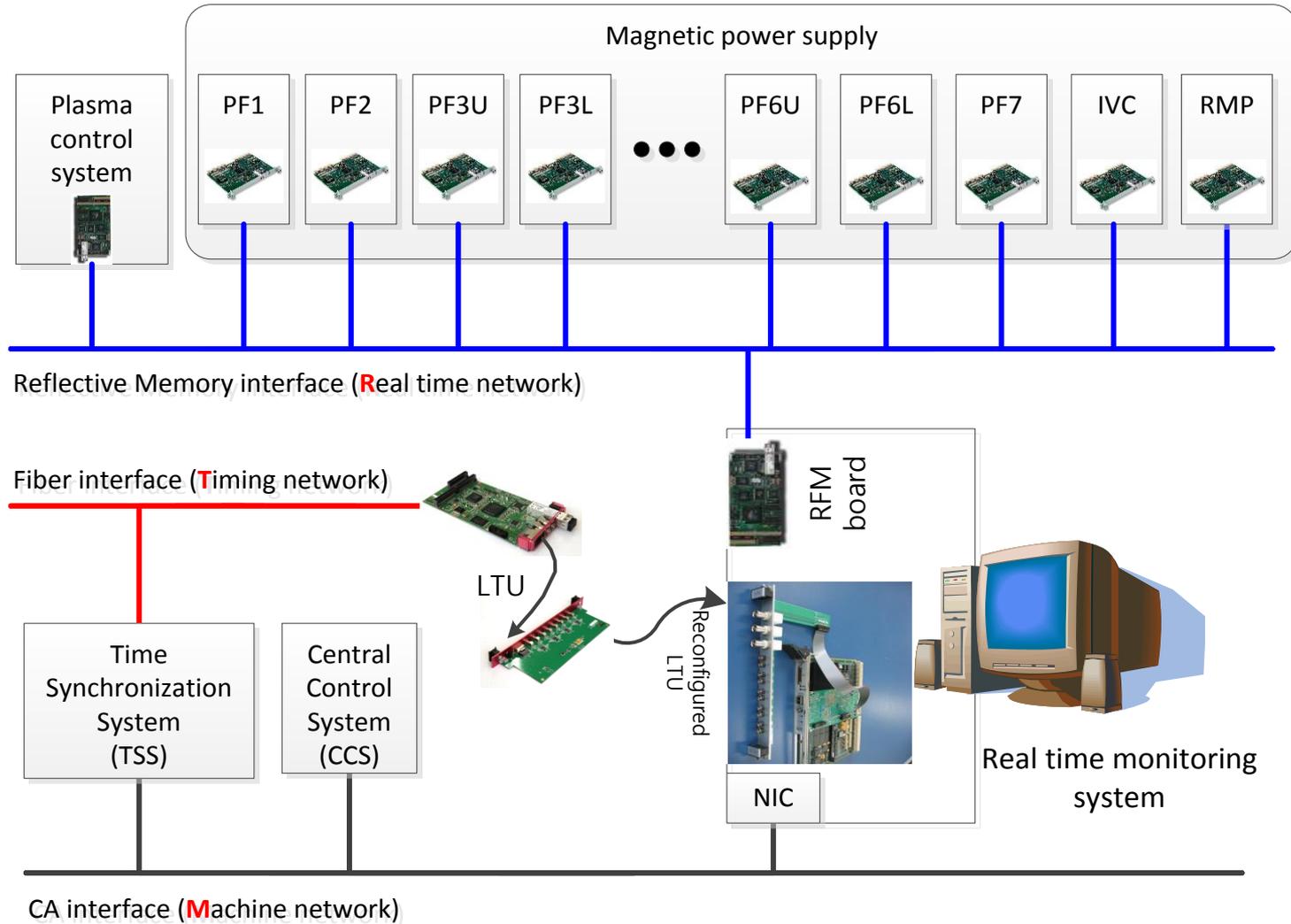
- ✓ MRG Realtime, TUNA

### ◆ Real-Time Support in EPICS

- ✓ Use POSIX priority scheduling by enabling special option
- ✓ Set CPU affinity using “epics-affinity-patch.txt”
  - From ITER and Cosylab.

# System configuration

## ● Simple layout



# System configuration

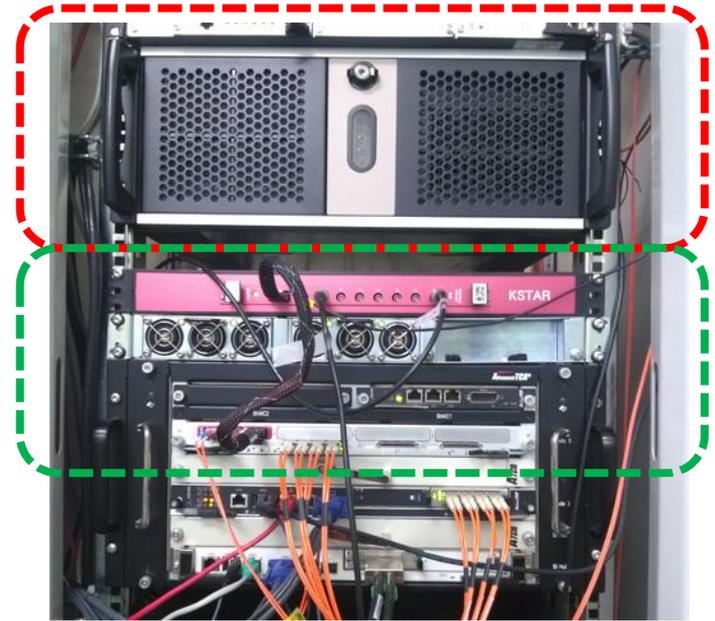
## ● Hardware features

### ◆ Host controller(red box)

- ✓ General rack mountable PC
- ✓ Intel® Core™ i7-3930K , 3.2GHz, 12MB cache
- ✓ 6 cores, no Turbo Boost, no Hyper-Threading
- ✓ DDR3 16GB
- ✓ 150GB SSD

### ◆ Local Timing Unit v.2(green box)

- ✓ Synchronized with Central Timing Unit (CTU) via fiber optics, 2Gbps
- ✓ Provide 10KHz clock to the event generator
- ✓ Generate reference clock for time measurement
- ✓ Installed on ATCA platform with PMC extension board



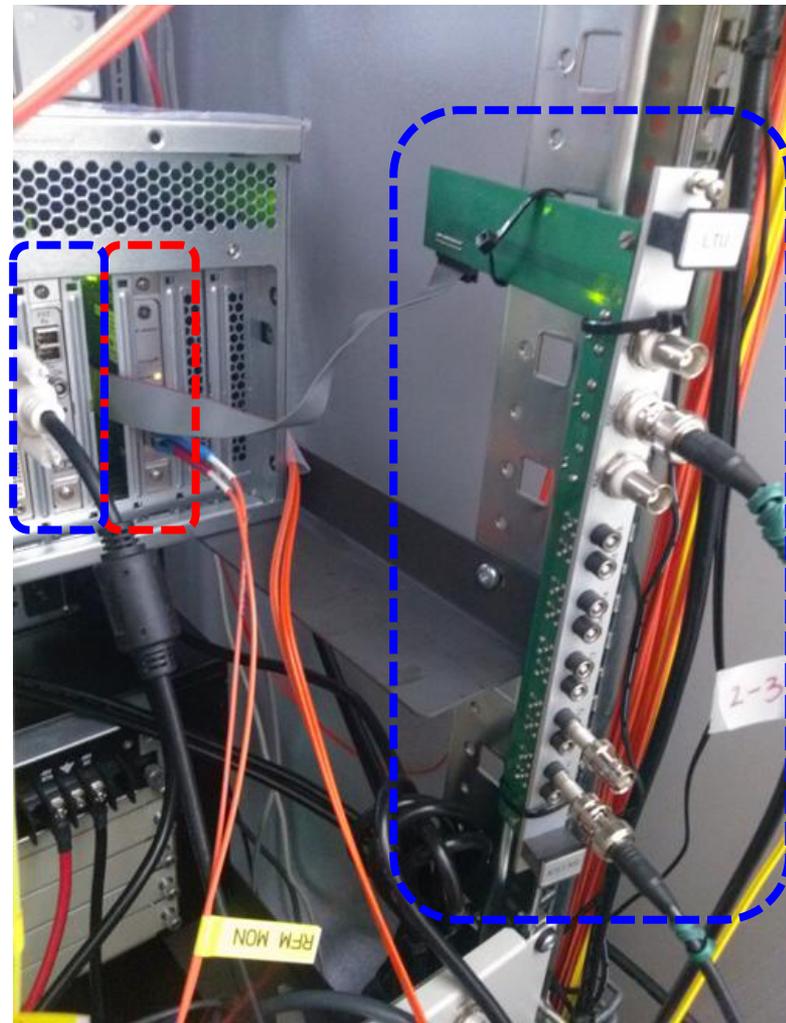
## ● Hardware features

### ◆ Event generator(blue box)

- ✓ Customized board using FPGA flexibility
- ✓ LTU v.1 was reconfigured event generator
- ✓ PMC true form-factor, Virtex-4 FX20
- ✓ 10 digital output for time checking

### ◆ Reflective Memory board(red box)

- ✓ High speed, easy to use fiber-optic network (2.12 Gbaud serially)
- ✓ Fiber network transfer rate 43 MByte/s to 170 MByte/s
- ✓ Star topology in KSTAR
- ✓ Main interface for feedback control between PCS and MPS



## ● Software features

### ◆ Host controller

- ✓ Red Hat Enterprise Linux 6.2 (2.6.32) x86\_64
- ✓ MRG Realtime – 2.6.33.9-rt31.74.el6rt.x86\_64
  - Disable all unnecessary service except network.
  - But system need more tuning

### ◆ Application features

- ✓ Recent stable released EPICS (3.14.12.2)
- ✓ Apply the real-time support patch
- ✓ Built on KSTAR standard software framework
  - Organized under driver/device support routine
  - Sequence for synchronized operation
  - State notification mechanism which is commonly used in DAQ and control system
  - Standard template for fast development
- ✓ Use ram-disk for fast archiving: 2.5 GB/s

# Operation flow

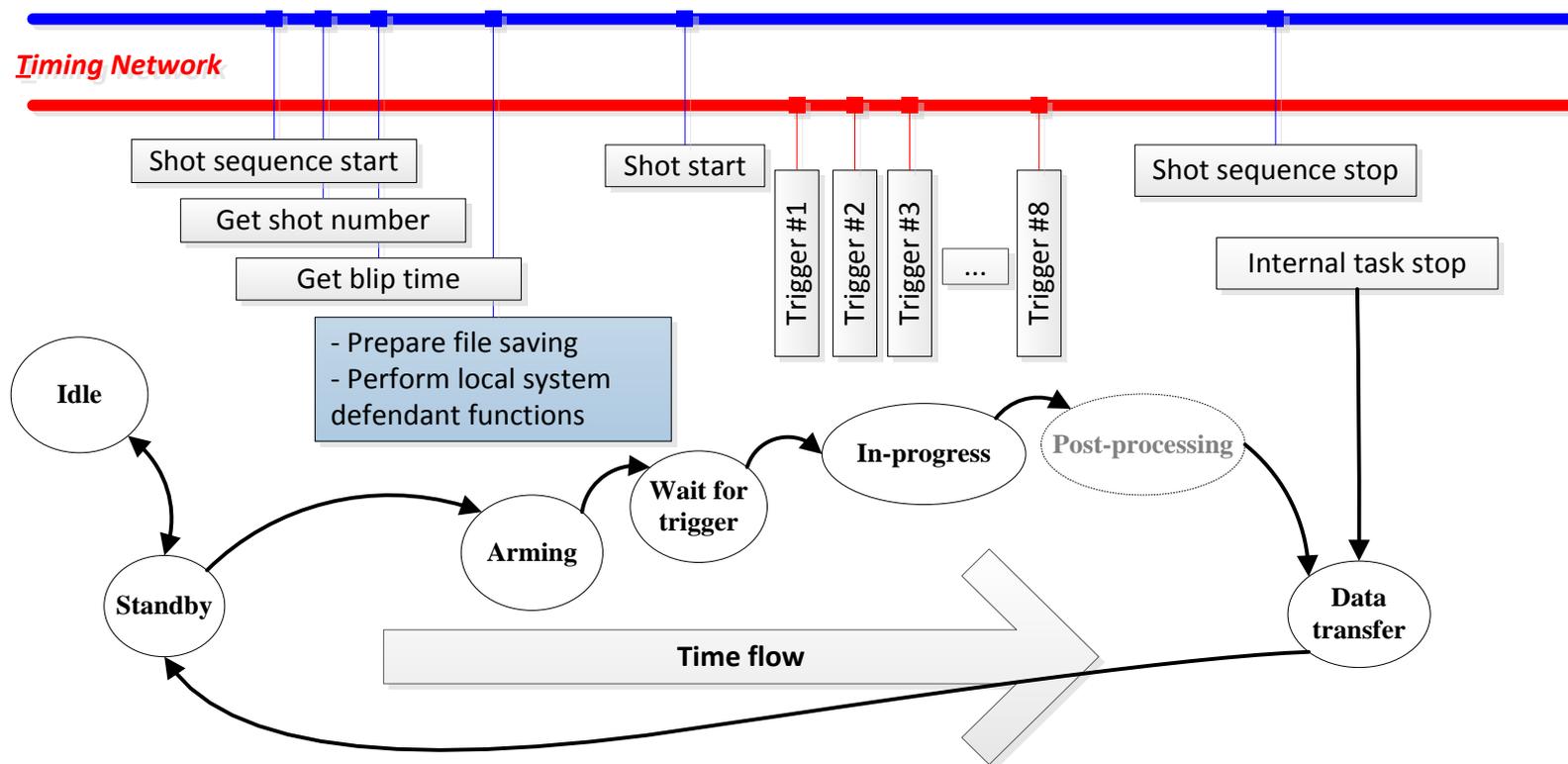
## ● KSTAR sequence synchronized operation

### ◆ Software framework

- ✓ It has been developed since 2009.
- ✓ Now we consider adding a real-time control feature

*Machine / Experiment Network*

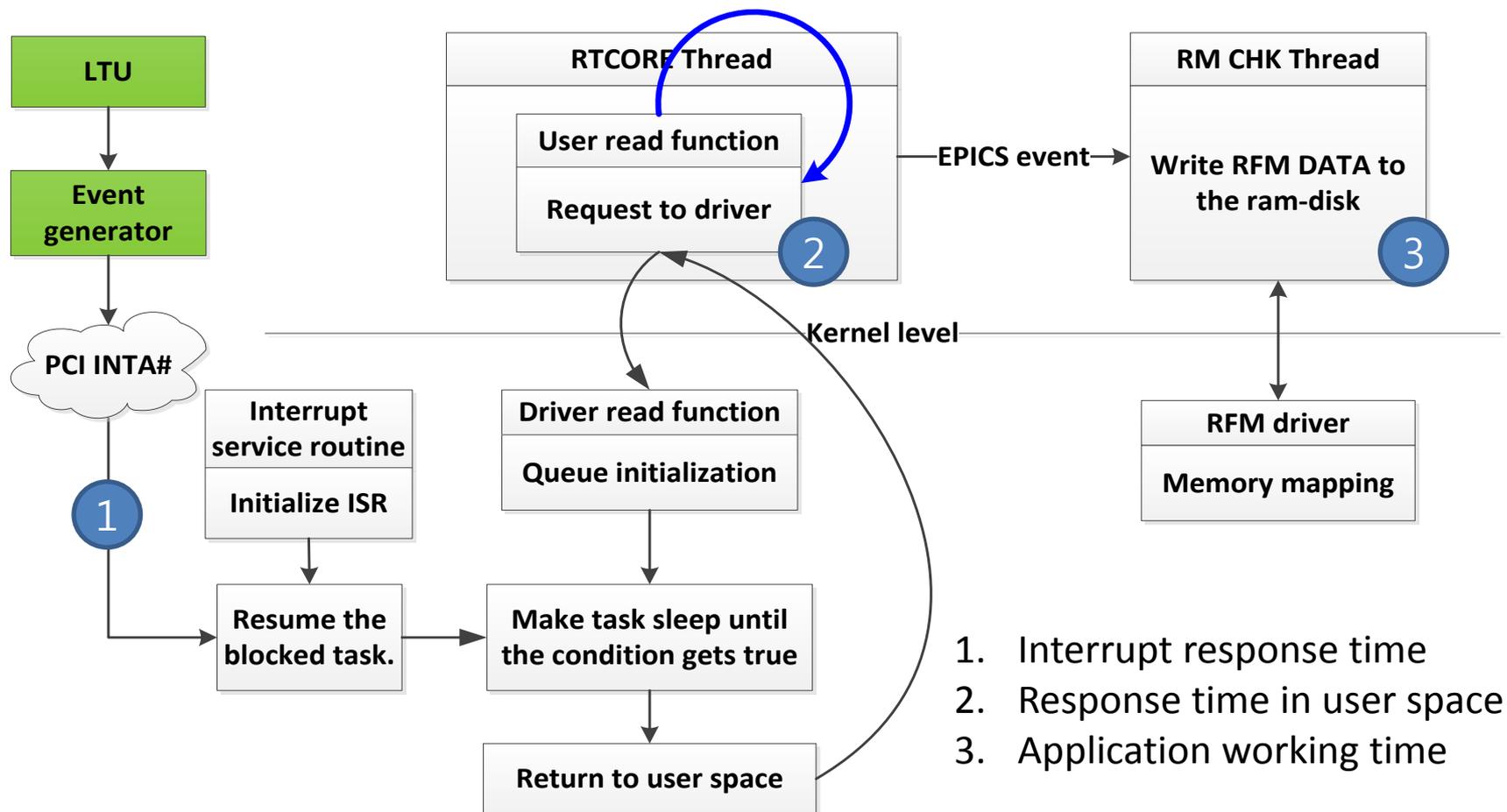
*Timing Network*



# Operation flow

## ● Application work flow

### ◆ Unit task per each input pulse



# Main function

## ● Archiving the real time data

### ◆ Text file contains these information

Event counter		Blip based	Self counter			PCS fault code	Force fast INTL	RMP P/S BI, MI, TI
RT counter	CHK counter	Current Time	PCS	All PFs	IVC			

IC. Over current	IP error	P/S fault	External fault	PF over voltage	PF voltage	PF over current	PF error	Ne error	IP minimum
------------------	----------	-----------	----------------	-----------------	------------	-----------------	----------	----------	------------

```

Real Time monitor - counter check.
PCS error - ICoc, IP, PS, EXT, PFov, PFvlt, PFoc, PF, NE, IPmin
RT clk: 10000, Hz: 10000Hz, step time: 0.000100
rtCnt daqCnt time(s) : PCS, PF1 PF2 (3U 3L) (4U 4L) (5U 5L) (6U 6L) PF7 IVC PCS_fault Fast_INTL |BI MI TI
1 1 -14.999900 : 29977 , 44302 33416 | 8965 40720 | 22920 21792 | 14116 3800 | 46171 37240 | 24805 31628 0000 H | 81f8 80fa 818b
2 2 -14.999800 : 29977 , 44303 33416 | 8966 40721 | 22920 21792 | 14116 3801 | 46172 37241 | 24805 31628 0000 H | 81f8 80fa 818b
3 3 -14.999700 : 29977 , 44303 33417 | 8966 40721 | 22921 21793 | 14117 3801 | 46172 37241 | 24806 31629 0000 H | 81f8 80fa 818b
4 4 -14.999600 : 29977 , 44304 33417 | 8967 40722 | 22921 21793 | 14117 3802 | 46173 37242 | 24806 31629 0000 H | 81f8 80fa 818b
5 5 -14.999500 : 29977 , 44304 33418 | 8967 40722 | 22922 21794 | 14118 3802 | 46173 37242 | 24807 31629 0000 H | 81f8 80fa 818b
6 6 -14.999400 : 29977 , 44305 33418 | 8968 40723 | 22922 21794 | 14118 3803 | 46174 37243 | 24807 31629 0000 H | 81f8 80fa 818b
7 7 -14.999300 : 29977 , 44305 33419 | 8968 40723 | 22923 21795 | 14119 3803 | 46174 37243 | 24808 31630 0000 H | 81f8 80fa 818b
8 8 -14.999200 : 29977 , 44306 33419 | 8969 40724 | 22923 21795 | 14119 3804 | 46175 37244 | 24808 31630 0000 H | 81f8 80fa 818b
9 9 -14.999100 : 29977 , 44306 33420 | 8969 40724 | 22924 21796 | 14120 3804 | 46175 37244 | 24809 31631 0000 H | 81f8 80fa 818b
10 10 -14.999000 : 29977 , 44307 33420 | 8970 40725 | 22924 21796 | 14120 3805 | 46176 37245 | 24809 31631 0000 H | 81f8 80fa 818b
11 11 -14.998900 : 29977 , 44307 33421 | 8970 40725 | 22925 21797 | 14121 3805 | 46176 37245 | 24810 31632 0000 H | 81f8 80fa 818b
  
```

# Main function

## ● RTMON' graphic user interface

- ✓ Developed by using QT based KWT
- ✓ Standard layout for Operation mode
- ✓ LTU configuration

The screenshot displays the RT-MON software interface. At the top, there are status tabs: Shot Preparation (highlighted), MPS Standby, PCS Standby, MPS Run, Shot Start, During Shot, End of Shot, and MPS Stop. Below these is a 'Shot Number' field containing '7314'. The main area is divided into several sections:

- Remote/Local Control:** A 'Remote' button is highlighted in green, with a 'Local' button below it.
- Configuration Tables:** Three tables for clock configurations (ITER PCS clk #1, Ref clk #2, and RTMON clk #8). Each table has columns for 'Rising', 'Falling', 'T0', 'T1', and 'Clock', with 'Setup' buttons.
- Parameter Settings:** Fields for 'RT Clock (Hz): 10000', 'Sampling rate (Hz): 10000', 'Start time (s): -15.0000', 'Run Period (s): 40.0000', 'Update RT PV: Disable', and 'Update 2nd RT PV: Disable'.
- PF Channels:** A row of 13 input fields with values: 13990, 2851, 28541, 11131, 42649, 42129, 33599, 24263, 16581, 6460, 44299, 14132, 30538. Below are labels for PF1(30c) through PCS(10c).
- RT Cnt and User Read offset:** Input fields for 'RT counter' (0), 'DAQ counter' (0), 'Offset' (0, 4, 8, c), and 'Value' (0, 196, 720896, 4294963122).
- Operatio mode (highlighted in red dashed box):** Contains a 'Remote' button (highlighted in green), a 'Local' button, 'MDSplus setup' (Create tree: 1.0), 'Control' (Release, Arming, Stop, Run, LTU stop, LTU start), and 'Status' (Standby, RTMON requires arming! 0x).

# System tuning

## ● Application binding

### ◆ Set CPU affinity

- ✓ Boot option *isolcpus=1-5*
- ✓ `epicsThreadSetCPUAffinity( pthreadInfo, "5")`
- ✓ `epicsThreadSetPosixPriority(pthreadInfo, 92, "SCHED_FIFO")`

```

var
epics> epicsThreadShowAll 1
      NAME          EPICS ID      PTHREAD ID    LWP ID    OSIPRI    OSSPRI    STATE    POLICY AFFINITY
      _main_         0x1a83190      0              0          0          0         OK       ? ?
      errlog         0x1a8cd50      140184245008128 6489       10         10        OK       SCHED_FIFO 0
      RTMON_ctrl     0x1a941b0      140184217679616 6490       10         10        OK       SCHED_FIFO 0
      RTMON_StopEvt  0x1a94480      140184217544448 6491       10         10        OK       SCHED_FIFO 0
      RTCORE_ctrl    0x1b39bd0      140184217409280 6492       10         10        OK       SCHED_FIFO 0
      RMCHK_ctrl     0x1b52960      140184217274112 6493       10         10        OK       SCHED_FIFO 0
      taskwd         0x1b57000      140184217138944 6494       10         10        OK       SCHED_FIFO 0
      timerQueue     0x1b64620      140184217003776 6495       70         69        OK       SCHED_FIFO 0
      cbLow          0x1b57190      140184216737536 6496       59         58        OK       SCHED_FIFO 0
      cbMedium       0x1b6cb20      140184216209152 6497       64         63        OK       SCHED_FIFO 0
      cbHigh         0x1b6ce10      140184215680768 6498       71         70        OK       SCHED_FIFO 0
      dbCaLink       0x1b6d2d0      140184215152384 6499       50         50        OK       SCHED_FIFO 0
      KSTAR_ShotSeq  0x1b6d560      140184214624000 6500       90         89        OK       SCHED_FIFO 0
      timerQueue     0x7f7f08010330 140184213829376 6502       92         91        OK       SCHED_FIFO 0
      CAC-UDP        0x7f7f08021f80 140184211248896 6503       94         93        OK       SCHED_FIFO 0
      ipToAsciiProxy 0x7f7f0800fbb0 140184214357760 6501       10         10        OK       SCHED_FIFO 0
      CAC-TCP-send   0x7f7f0c00a250 140184210454272 6505       92         91        OK       SCHED_FIFO 0
      CAC-TCP-recv   0x7f7f0c009ea0 140184210982656 6504       88         87        OK       SCHED_FIFO 0
      CAC-TCP-send   0x7f7f0c017360 140184209131264 6507       92         91        OK       SCHED_FIFO 0
      CAC-TCP-recv   0x7f7f0c023760 140184209059040 6506       88         87        OK       SCHED_FIFO 0
      RMCHK_DAQ      0x1b69e50      140184206583552 6508       93         92        OK       SCHED_FIFO 5
      RTCORE_RT      0x1b6a170      140184206317312 6509       93         92        OK       SCHED_FIFO 5
      scanOnce       0x1b80260      140183305534112 6510       70         69        OK       SCHED_FIFO 0
      scan10         0x1b81270      140184206051072 6511       60         59        OK       SCHED_FIFO 0
  
```

# System tuning

## ● Interrupt binding

### ◆ Set IRQ attribute by Tuna

✓ Priority 95, Affinity 5, Policy FIFO

**Set IRQ Attributes**

Set attributes for this IRQ:  
IRQ 16 (PID 5206), pri 95, aff 5, intLTU.0

Policy: SCHED\_FIFO Scheduler priority: 95 Affinity: 5

Buttons: Cancel, OK

**Tuna**

Filter	CPU	Usage	IRQ	PID	Policy	Priority	Affinity	Events	Users
<input checked="" type="checkbox"/>	0	0	0	-1	-1	-1	0-5	2159	timer
<input checked="" type="checkbox"/>	2	0	1	754	FIFO	85	0-4	8	i8042
<input checked="" type="checkbox"/>	1	4	8	765	FIFO	85	0-4	21	rtc0
<input checked="" type="checkbox"/>	3	0	9	225	FIFO	85	0-4	0	acpi
<input checked="" type="checkbox"/>	4	0	16	5206	FIFO	95	5	525576919	intLTU.0
<input checked="" type="checkbox"/>	5	0	17	1767	FIFO	85	0-4	9453	firewire_ohci
			23	1670	FIFO	85	0-4	92	ehci hcd:usb1 ehci hcd:usb2

**cat /proc/interrupts**

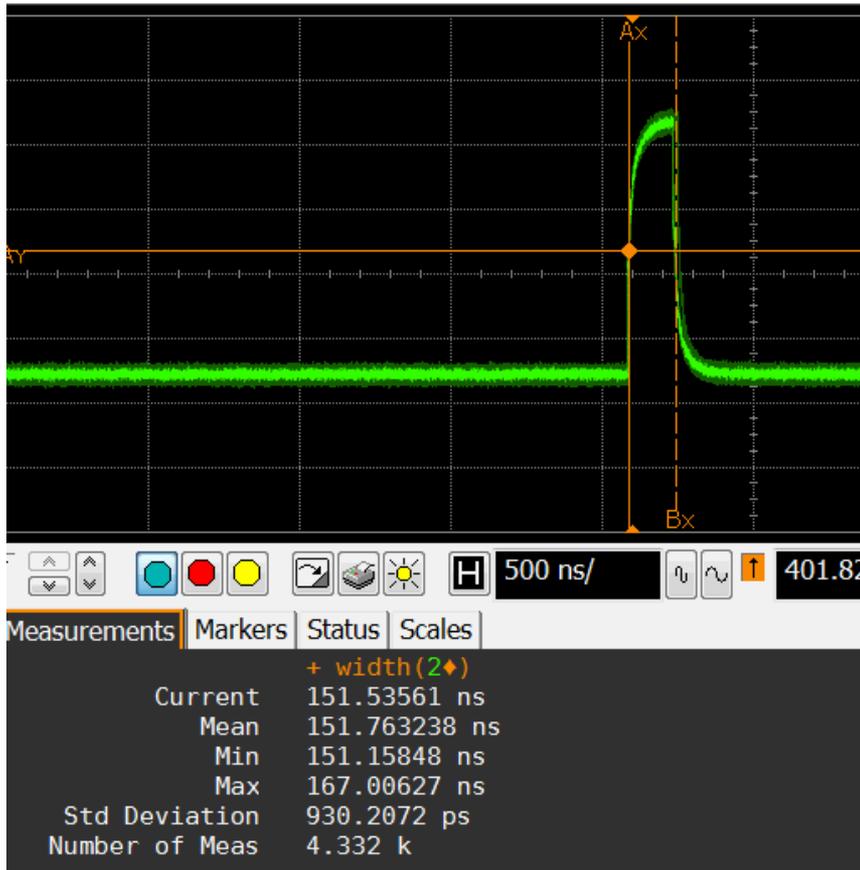
```

CPU0      CPU1      CPU2      CPU3      CPU4      CPU5
0:         142       2004      0          0          0          IO-APIC-edge timer
1:         0         8         0          0          0          0          IO-APIC-edge i8042
8:         0         21        0          0          0          0          IO-APIC-edge rtc0
9:         0         0         0          2          0          512479144 IO-APIC-fasteoi intLTU.0
16:        0         0         0          0          0          0          IO-APIC-fasteoi firewire_ohci
17:        0         0         0          9391       0          0          IO-APIC-fasteoi ehci hcd:usb1, ehci hcd:usb2
23:        92         0         0          0          0          0          IO-APIC-fasteoi PCI RFM2g
40:        0         0         0          0          0          0          PCI-MSI-edge aerdrv
48:        0         0         0          0          0          0          PCI-MSI-edge aerdrv
49:        0         0         0          0          0          0          PCI-MSI-edge aerdrv
50:        0         0         0          0          0          0          PCI-MSI-edge aerdrv
5206      FIFO      95        5          510697311 2          irq/16-intLTU.0
5442      OTHER    0         0-4        19069      2410      sshd: root@pts/4
5446      OTHER    0         0-4        325        67         -bash
5797      OTHER    0         0-4        107        467        ../bin/linux-x86_64/RTMON ./s
5841      OTHER    0         0-4        2272      99         sshd: root@pts/5
5846      OTHER    0         0-4        569        230        -bash
6317      OTHER    0         0-4        8360      5010      sshd: root@pts/6
6321      OTHER    0         0-4        217        502        -bash
6408      OTHER    0         0-4        3625      9619      /usr/bin/python /usr/bin/tuna
6409      OTHER    0         0-5        9          2          flush-253:0
  
```

## ● Accuracy

### ◆ Oscilloscope

✓ Model-DSO9404A , Infiniium DSO - 4 GHz, 10/20 GSa/s, 4 Ch



- To measure the consumed time, we use TTL output channel on the event generator
- Two command to make single pulse.

```
#1 WRITE32(pRTcore->base0 + 0x4, 0x1);  
#2 WRITE32(pRTcore->base0 + 0x4, 0x0);
```

- Takes about 151 nanoseconds

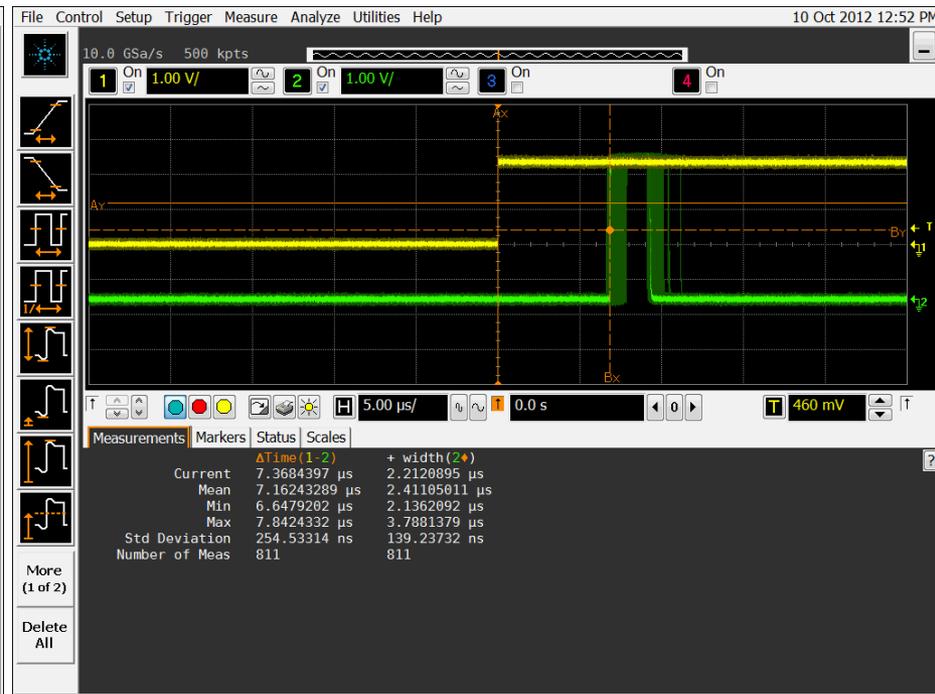
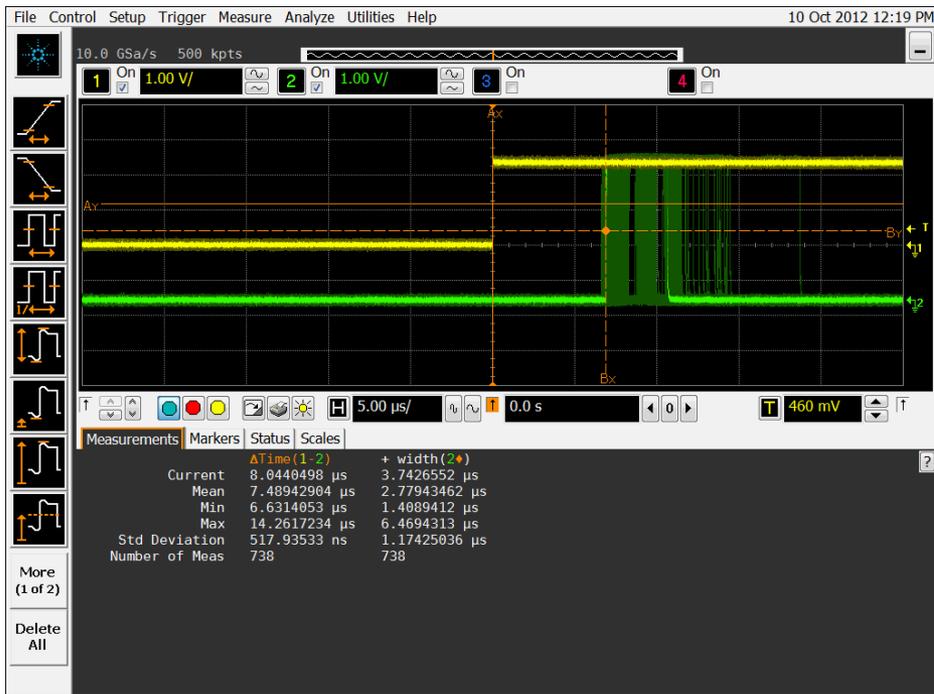
# Test results

## ● Captured test images

### ◆ Interrupt response time in kernel space

■ Not use CPU affinity

■ Use CPU affinity



# Test results

## ● Measured time value

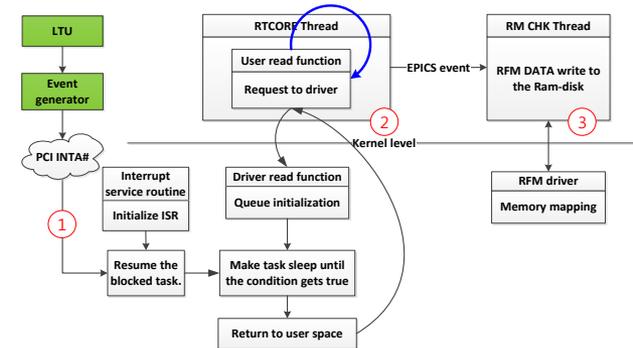
### ◆ Condition

- ✓ 10KHz interrupt in normal operation
- ✓ unit=microsecond, doing in a few minutes

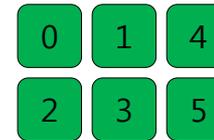
case	mean	min	max	jitter	Check point
1	7.489	6.6314	14.2617	7.6303	LTU → Kernel ISR
2	7.162	6.647	7.842	1.195	
3	6.919	6.409	8.199	1.789	

case	mean	min	max	jitter	Check point
1	10.369	8.876	14.473	5.596	LTU → User space return time
2	9.679	9.249	10.99	1.74	
3	8.567	8.104	10.403	2.298	

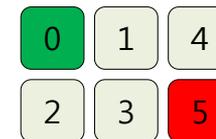
case	mean	min	max	jitter	Check point
1	40.588	38.804	47.873	9.068	LTU → final action include user application
2	38.932	38.1	43.259	5.159	
3	38.831	37.819	42.236	4.417	



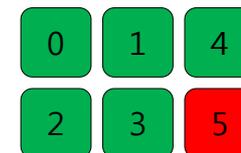
### 1. MRGR, no CPU affinity



### 2. MRGR, isolcpus=1-5, affinity=5



### 3. MRGR, isolcpus=5, affinity=5



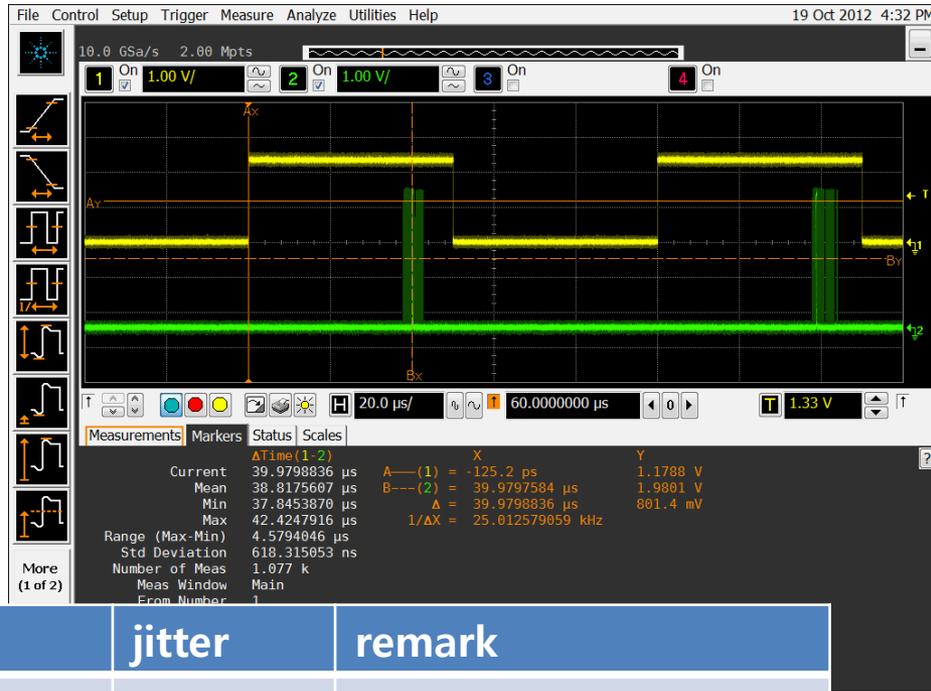
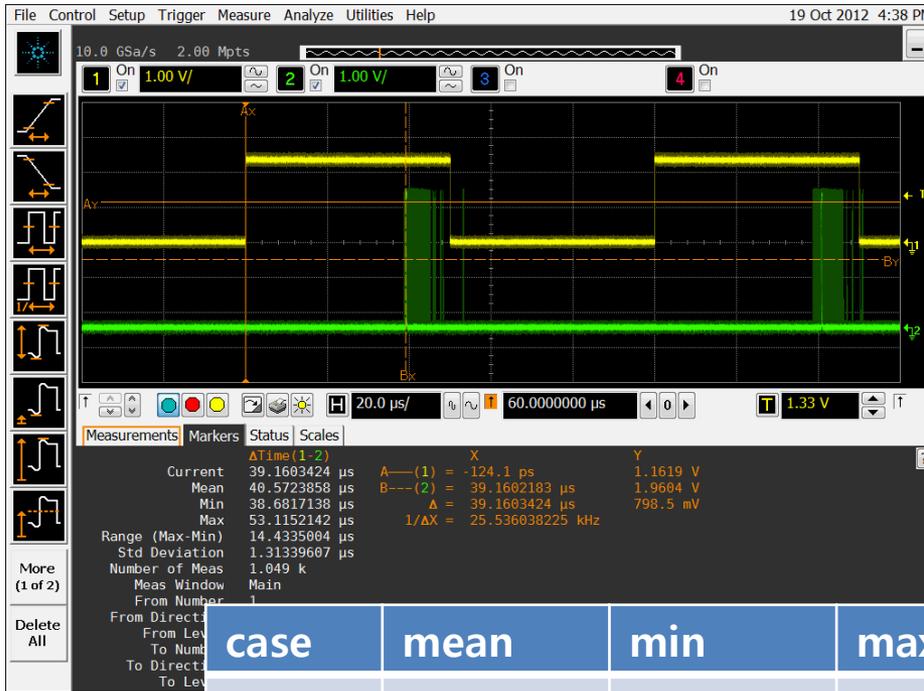
# Test results

## ● Special case

### ◆ Total time with heavy job

- Not use CPU affinity

- Use CPU affinity



case	mean	min	max	jitter	remark
1	40.572	38.681	53.115	14.433	LTU → final action with heavy job
2	38.817	37.845	42.424	4.579	
3	38.817	37.845	42.424	4.579	

# Conclusion and Future work

## ● Conclusion

### ◆ Achieved 10KHz stable operation

- ✓ file I/O, network connection(CA) are alive
- ✓ Possible in normal or harsh condition

### ◆ Possible 20KHz operation

- ✓ 20KHz (50 us) operation also possible
- ✓ It depends on target application

### ◆ Effect of CPU affinity

- ✓ Dominant effect is using RTOS (MRGR)
- ✓ Using CPU affinity increase the system stability

## ● Future work

### ◆ Evaluation in feedback control system

- ✓ Develop real time feedback control system on plasma density
- ✓ Survey of real time network performance

**Thanks for your attention.**