

ROD TestStand Overview

LukasTomasek <tomasekl@fzu.cz>

LBL, FZU ASCR Prague

Test Stand basics

Purpose

- initial testing and debugging a standalone ROD,
- production testing and ROD maintenance.

Software

- programming language ANSI C
- MS Windows NT (2000, 9X) operating system
- NI LabWindows/CVI 5.5 development software

Hardware

- PC with NI PCI-MXI-2 interface card,
- VME crate with NI VME-MXI-2 card

Main TestStand tasks and features

- **ROD initialization and configuration:**
 - loading program for FPGAs to flash memories,
 - R/W FPGA command/status registers (board reset...),
 - loading program for DSPs (MDSP flash, slaves directly),
- **ROD control:**
 - by *PrimitiveLists*,
 - “fast” commands (*VmeCommandRegister* bits);

Any Rod task is defined as a sequence of basic commands - *CommandList*, which can be edited and saved in the interactive *CommandListEditor*.

The implementation and the structure of the *CommandList* is very similar to the *PrimitiveList*.
=> TestStand is ready to run the most of possible user defined tasks **without recompiling the code**.

- **ROD monitoring:**
 - periodic checking of the ROD status registers (ControllerFPGA, MDSP),
 - readout text messages from the *TextBuffers*.

TestStand supports one VME crate - up to 16 Rods can run simultaneously.

CommandList Editor

COMMAND LIST EDITOR

Main! PrimListEditor! DataFileCreator! HideWindow!

COMMAND

CmdRepetitions Continue if error?

3 - Send PrimList

- 0 - R/W VME
- 1 - R/W Master
- 2 - R/W Slave
- ✓ 3 - Send PrimList
- 4 - CompareFiles
- 5 - BusTest
- 6 - RunChildCommandList
- 7 - R/W Fpga Flash
- 8 - Write to MDSP Flash
- 9 - BOC Setup
- A - Run External Program
- B - Delay

Slot#

ALL

SEND TO ROD

ADD BEFORE >

REPLACE >>

ADD AFTER >>

<<EDIT CMD

DELETE CMD

LOAD LIST

INSERT LIST

SAVE LIST

DELETE LIST

COMMAND LIST

CmdCount **14** LastFile

c:\RodTS\Data\Testing\fifoTesting\verifyAllFifos.clst

ID	COMMAND	Repeat
1	3 - Send PrimList	0
2	3 - Send PrimList	0
3	3 - Send PrimList	0
4	3 - Send PrimList	0
5	3 - Send PrimList	0
6	3 - Send PrimList	0
7	4 - CompareFiles	0
8	4 - CompareFiles	0
9	4 - CompareFiles	0
10	4 - CompareFiles	0
11	4 - CompareFiles	0
12	4 - CompareFiles	0
13	4 - CompareFiles	0
14	4 - CompareFiles	0

COMMAND LIST EDITOR

Main! PrimListEditor! DataFileCreator! HideWindow!

COMMAND

CmdRepetitions Continue if error?

3 - Send PrimList

INPUT DATA

Repeat

EXECUTION
 BUILDING

PrimList

c:\RodTS\Data\Testing\fifoTesting\writeReadDbgFifoRandom.mlist

Slot#

ALL

COMMAND LIST

CmdCount **14** LastFile

c:\RodTS\Data\Testing\fifoTesting\verifyAllFifos.clst

ID	COMMAND	Repeat
1	3 - Send PrimList	0
2	3 - Send PrimList	0
3	3 - Send PrimList	0
4	3 - Send PrimList	0
5	3 - Send PrimList	0
6	3 - Send PrimList	0
7	4 - CompareFiles	0
8	4 - CompareFiles	0
9	4 - CompareFiles	0
10	4 - CompareFiles	0
11	4 - CompareFiles	0
12	4 - CompareFiles	0
13	4 - CompareFiles	0
14	4 - CompareFiles	0

Primitive List Editor

PRIM LIST EDITOR
Main! CommandListEditor! DataFileCreator! HideWindow!

PRIMITIVE
MASTER ▾ Slot# **ALL** ▾

2006 - RW_FIFO **SEND TO ROD**

0 - ECHO
1 - SET_ERRMSG_MASK
2 - PAUSE_LIST
3 - EVENT_TRAP_SETUP
4 - SET_MEMORY
5 - COPY_MEMORY
6 - MEMORY_TEST
7 - SET_LED
8 - FLASH_LED
9 - SEND_DATA
A - MODULE_MASK
B - SET_TRIGGER
C - START_TASK
D - TASK_OPERATION
E - TEST
F - WRITE_BUFFER
2000 - RW_SLAVE_MEMORY
2001 - TRANS_SERIAL_DATA
2002 - START_SLAVE_EXECUTING
2003 - CONFIG_SLAVE
2004 - RW_REG_FIELD
2005 - POLL_REG_FIELD
✓ 2006 - RW_FIFO
2007 - >>SEND_SLAVE_LIST
2008 - START_SLAVE_LIST
2009 - SLAVE_LIST_OP
200A - BUILD_STREAM
200B - SEND_STREAM
200C - RW_MODULE_DATA
200D - SEND_CONFIG

MASTER LIST
ListLength PrimCount AllPrimCount
PrimList x 6084 14 14
RepList x 6042 12 LastFile

S:\Data\Testing\fifoTesting\writeReadDbgFifoRandom.mlist

ID	PRIMITIVE	PrimLn	RepLn
1	2004 - RW_REG_FIELD	9	0
2	2004 - RW_REG_FIELD	9	0
3	2005 - POLL_REG_FIELD	9	5
4	2005 - POLL_REG_FIELD	9	5
5	2005 - POLL_REG_FIELD	9	5
6	2005 - POLL_REG_FIELD	9	5
7	2006 - RW_FIFO	1809	5
8	2006 - RW_FIFO	1809	5
9	2006 - RW_FIFO	1809	5
10	2006 - RW_FIFO	1809	5
11	2006 - RW_FIFO	9	1805
12	2006 - RW_FIFO	9	1805
13	2006 - RW_FIFO	9	1805
14	2006 - RW_FIFO	9	1805

ADD BEFORE>> Ctrl+
REPLACE>> <->
ADD AFTER>> >>
ViewPrim ↑↓ Ctrl+
<<EDIT PRIM<-> Ctrl+
DELETE PRIM Del
LOAD LIST
INSERT LIST
SAVE LIST
DELETE LIST

PRIM LIST EDITOR
 Main! CommandListEditor! DataFileCreator! HideWindow!

PRIMITIVE
 MASTER
 2006 - RW_FIFO

Slot# ALL
 SEND TO ROD

MASTER LIST
 ListLength PrimCount AllPrimCount
 PrimList x 6084 14 14
 RepList x 6042 12 LastFile

S:\Data\Testing\fifoTesting\writeReadDbgFifoRandom.mlist

ID	PRIMITIVE	PrimLn	RepLn
1	2004 - RW_REG_FIELD	9	0
2	2004 - RW_REG_FIELD	9	0
3	2005 - POLL_REG_FIELD	9	5
4	2005 - POLL_REG_FIELD	9	5
5	2005 - POLL_REG_FIELD	9	5
6	2005 - POLL_REG_FIELD	9	5
7	2006 - RW_FIFO	1809	5
8	2006 - RW_FIFO	1809	5
9	2006 - RW_FIFO	1809	5
10	2006 - RW_FIFO	1809	5
11	2006 - RW_FIFO	9	1805
12	2006 - RW_FIFO	9	1805
13	2006 - RW_FIFO	9	1805
14	2006 - RW_FIFO	9	1805

INPUT DATA

READ
WRITE

FifoID&Bank
 INPUT_MEM - A

FifoSize [elems]
 Input x 1000 48b
 Debug x 1000 48b
 EventAB x 4000 48b
 EventC x 00FF 32b
 Tim x 2000 8b

NumElements
 x 00001000

File Size
 x 00006000 B

INPUT FILE
 c:\RodTS\Data\Testing\fifoTesting\4Krandom1.bin

OUTPUT DATA

OUTPUT FILE SAVE OUTPUT DATA TO FILE?

ADD BEFORE
 REPLACE
 ADD AFTER
 ViewPrim
 <<EDIT PRIM
 DELETE PRIM
 LOAD LIST
 INSERT LIST
 SAVE LIST
 DELETE LIST

ROD Status/Control Panel (old version)

STATUS/CONTROL

Slot# 17 DMA priority HOST DSP CPU **ENABLED** Rodnit VME

PrimListBuild ListHandler RepListProcess

Idle Busy Prim List Built Error

Write Prim List Poll DspAckClr Read Rep List Error

Idle Busy Error

Prim List Built RepProc Busy

ErrorBuff InfoBuff DiagBuff XferBuff

Idle RdRqSet Readout Error

Idle RdRqSet Readout Error

Idle RdRqSet Readout Error

Idle RdRqSet Readout Error

ProcBusy ProcBusy ProcBusy ProcBusy

Status0 Command0

00000011 00000000

DspRunning 0 3 Abort OFF

Busy 1 2 Resume OFF

Executing 2 2 Resume OFF

Paused 3 1 Pause OFF

OutListRdy 4

DspAck 5 0 InListRdy

DmaAck A A DmaReq OFF

DmaErr 8

ErrorBuffNE 6 4 RdRq DIS

InfoBuffNE 7 5 RdRq DIS

DiagBuffNE 8 6 RdRq DIS

XferBuffNE 9 7 RdRq DIS

Status1 List Index

00000030 00000000

Read Stat Regs

Control Thread

Idle RodNit Busy

RunCmdList

Reset Resume

ListFIFO Counter

0 Reset 10

Repetitions Counter

0 Reset EXEC BUILD

SaveLists OFF

ListOverlap OFF

ChecksumMaster ON

ChecksumSlave ON

SEND PRIM LIST

ROD Controller Status

ROD S/N Type

1 SCT

SignalMux

DataPath TestPath

FE CMD OUT Enabled

MDSP-SP0 x FFFF FFFFFFFF

MDSP-SP1 x 0000 00000000

FE CMD Pulse Counter d 0

FE Occ Counters All Zero

Trigger Signal Decoder

Formatter ModeBits Encoder

FIFO A EF FF

FIFO B EF FF

EFB DynMask Encoder

FIFO EF FF

EVID Empty Error

Test Bench I/O Run

Debug FIFO I/O

Router/DSP Trap Status

En	Trap0	On	Trap1	On	WC
0	NONE		NONE		x 000
1	NONE		NONE		x 000
2	NONE		NONE		x 000
3	NONE		NONE		x 000

Header/Trailer Limit A B

ROD Busy A B

FIFO Status

FMB/EFBDM En EventMem

CorrTrig En A EF FF

Debug InMem B EF FF

Int TIM En

Read ROD Controller Status

COMMAND LIST STATUS

ClearStatusWindow! OpenStatusFile! CmdListEditor! Hide!

COMMAND LIST

NumCmds CmdIndex

43 0

ListRep Counter

Reset 0 0

If Error Default- Stop- Continue-

UpdateDisplay

0 - RAW VME

CmdRep Counter

Reset 0 0

S:\Data\Testing\loadSlaveCode\load&startAllSlave.clst

EDIT SEND COMMAND LIST

UpdateStatus Status

***** Date:07-15-2002, Time:20:28:19 *****

c:\RodTS\data\Testing\loadSlaveCode\load&startAllSlave.clst

CLIST START (NumCmds:43; ListRep:0)

CLIST ENT (CmdIndex:43; ListRepCount:1)

***** Time:20:28:22 *****

ERR MASK

ListIndex x 00000000

Time

ErrmsgMask x 00000000

RW REG FIELD

ListIndex Time

ObjectId Offset

x 00000000 x 00000000

Width DataOut

x 00000000 x 00000000

POLL REG FIELD

ListIndex Time

ObjectId Offset

x 00000000 x 00000000

Width DesiredValue

x 00000000 x 00000000

NotFound

MEMORY TEST

ListIndex Time

StartAddr MemorySize

x 00000000 x 00000000

ReturnCode x 0

HISTOGRAM SETUP

Slv	ctrBase	pulseBase
0	00000000	00000000
1	00000000	00000000
2	00000000	00000000
3	00000000	00000000

Capture FE Occ&OutFifoWC

FE OccCnt OutFifoWC

Link d 0 SaveData

Samples Capture

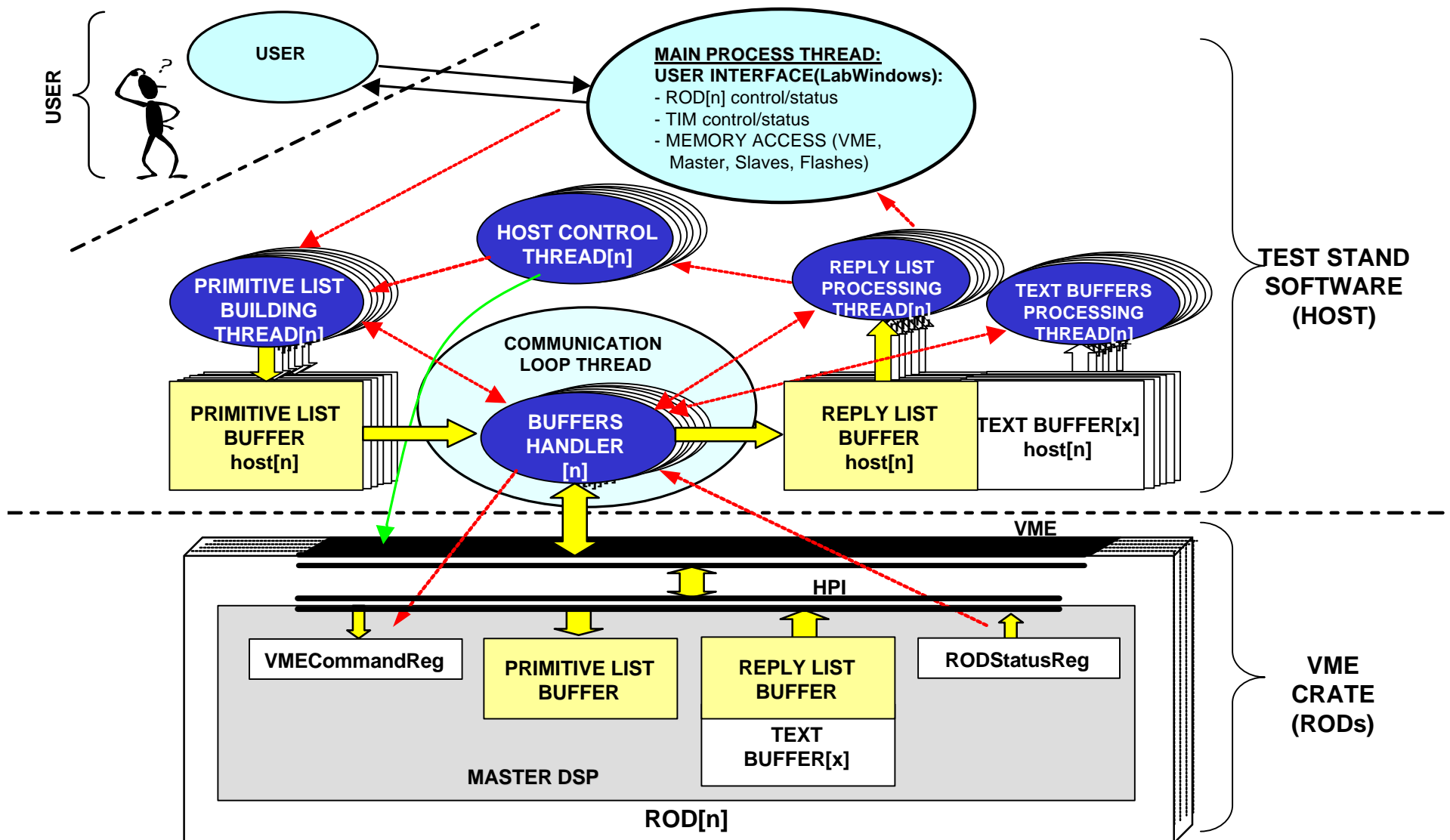
x 00100000 OFF

Overflow

SamplTime d 0.000 s

Total Time d 0.000 s

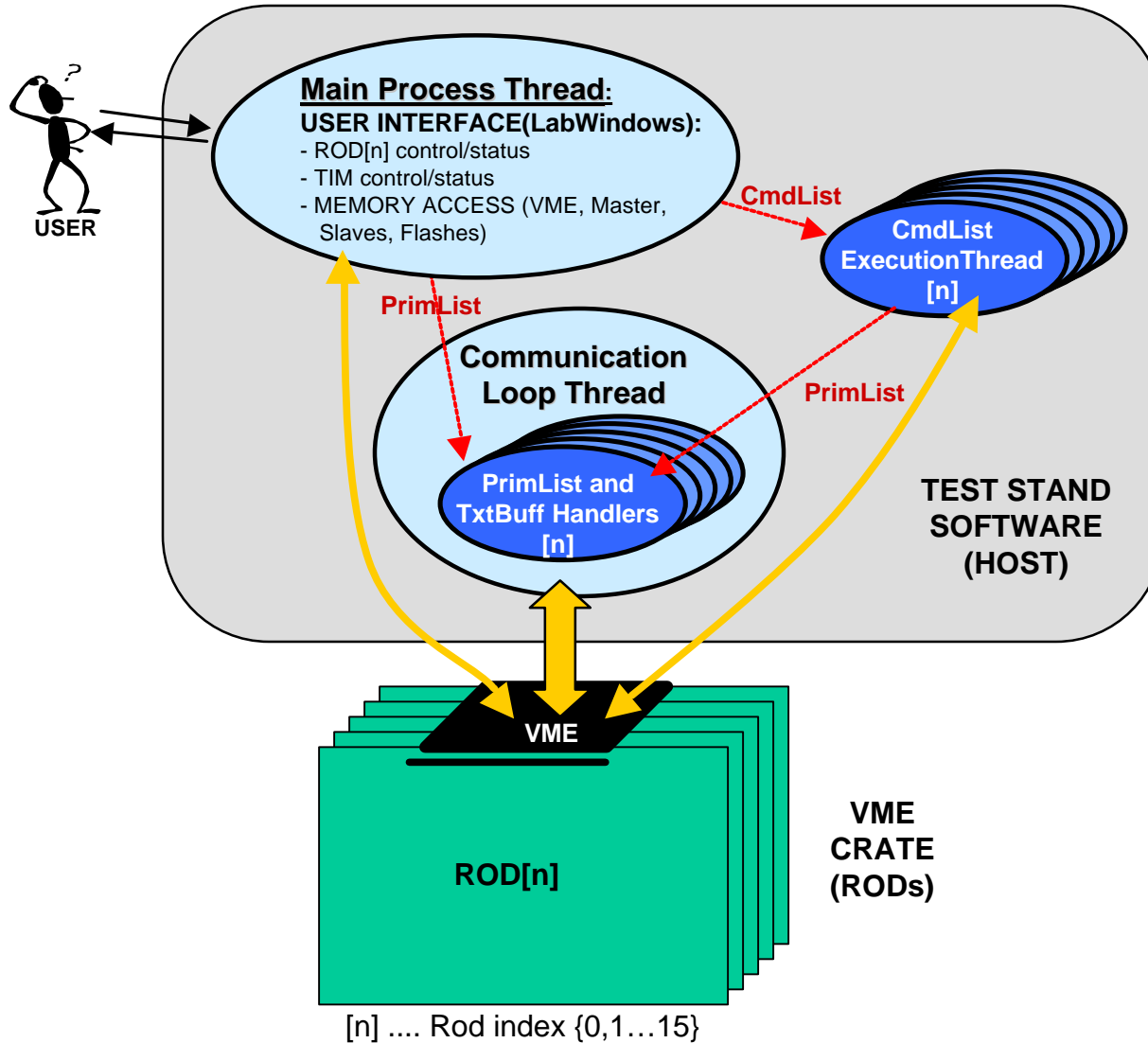
“Old” Test Stand Structure - Threads



Legend:

- Main data path
- Handshake/synchronization signals
- [n]** - Rod/Host index {0,1...15}

New "simpler" Test Stand Structure - Threads



Software design

- *single process multithreaded* application

Multithreading is necessary mainly because we need a “**responsive**” **user interface**, it’s also simpler, safer and easier to implement some functionality inside separate threads than inside the main program path (e.g. *ChildCmdList* execution).

The drawback is a difficult debugging, relatively complicated synchronization between the threads and mainly the portability problems to other systems.

=> The old TestStand architecture has been **simplified**, number of threads reduced to the useful minimum (*PrimListBuilding* and *ReplyListProcessing* moved to *PrimListHandler*, *TextBufferProcessing* to *TextBufferHandler*, the state machines were minimized, all unimportant program options removed, the user interface simplified).

Threads

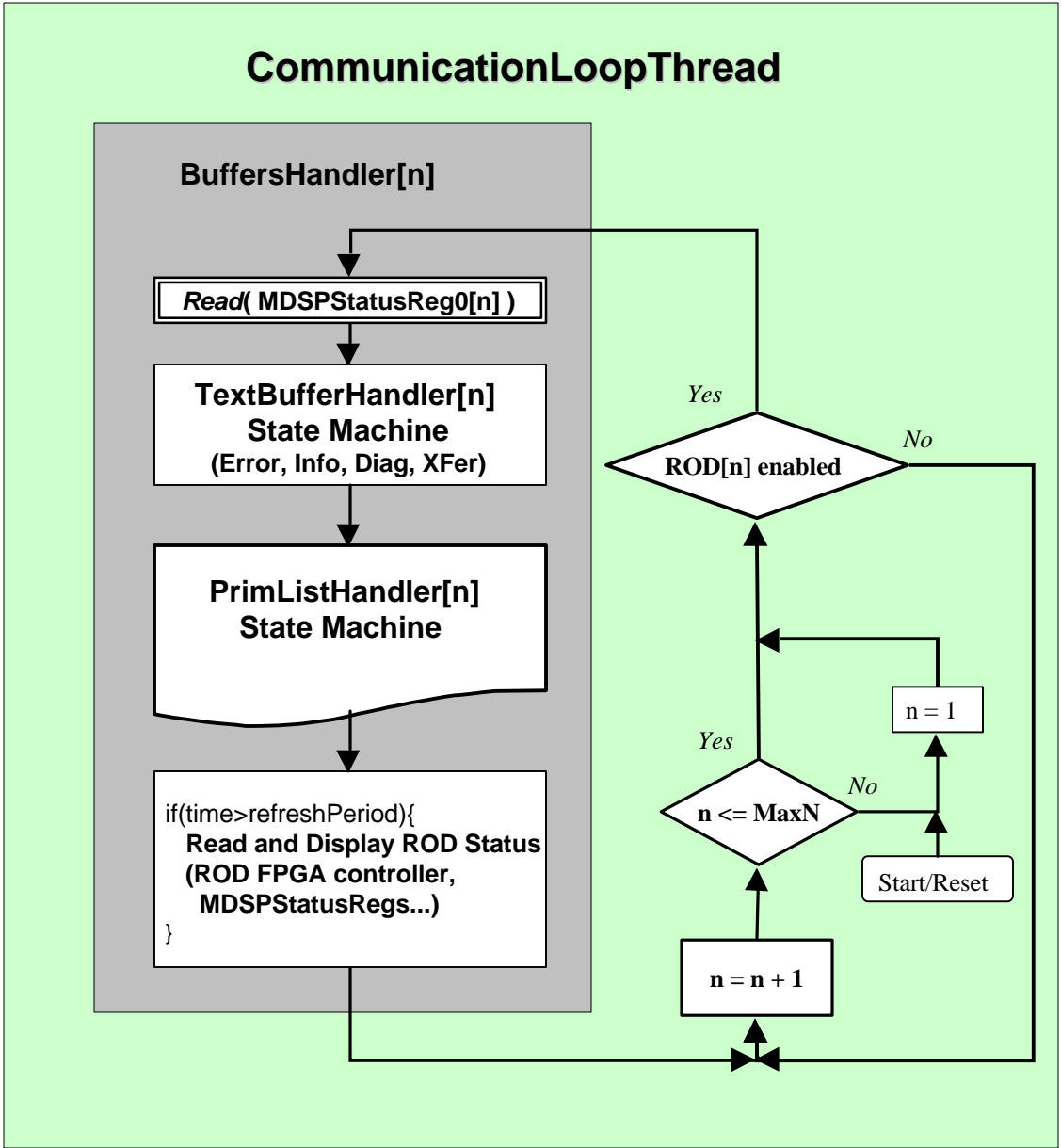
- **Main Thread**

- **interaction with the User** (LabWindows GUI),
 - Main Panel (program options, thread priorities...),
 - RW Memory Panels(VME, MDSP, SDSs, Flashes, FPGA Control/Status),
 - *PrimList* Editor, *CommandList* Editor,
 - ROD Status/Control Panel;

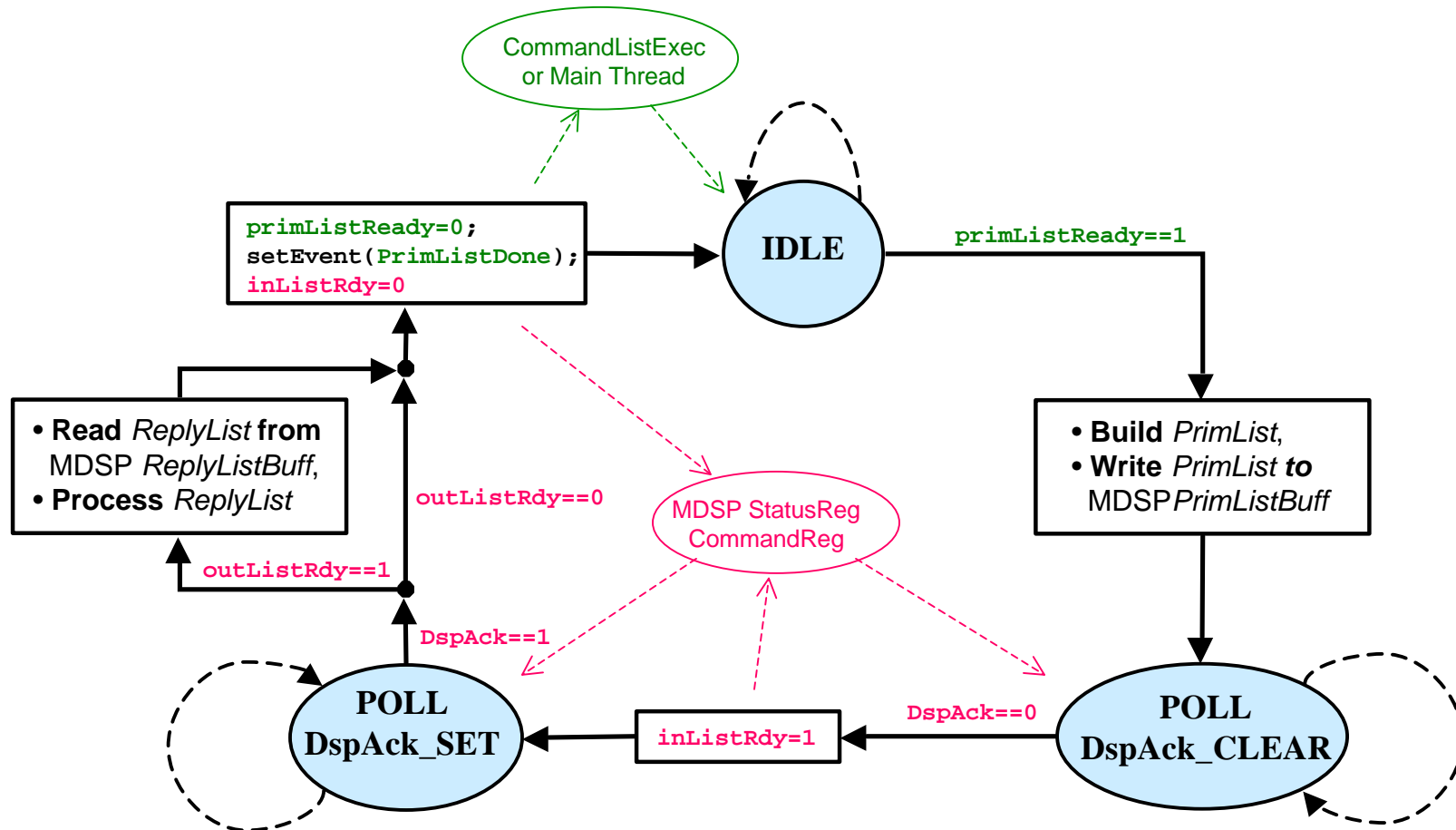
- **CmdListExecutionThread[n]** - *CommandList* execution.

- **CommunicationLoop**

- common to all RODs,
- distributes an access to the shared VME bus between all RODs in the crate on “*round robin*” basis,
- executes *PrimitiveLists* and processes *TextBuffer* messages;
- periodically monitors the status of all Rods in the crate.

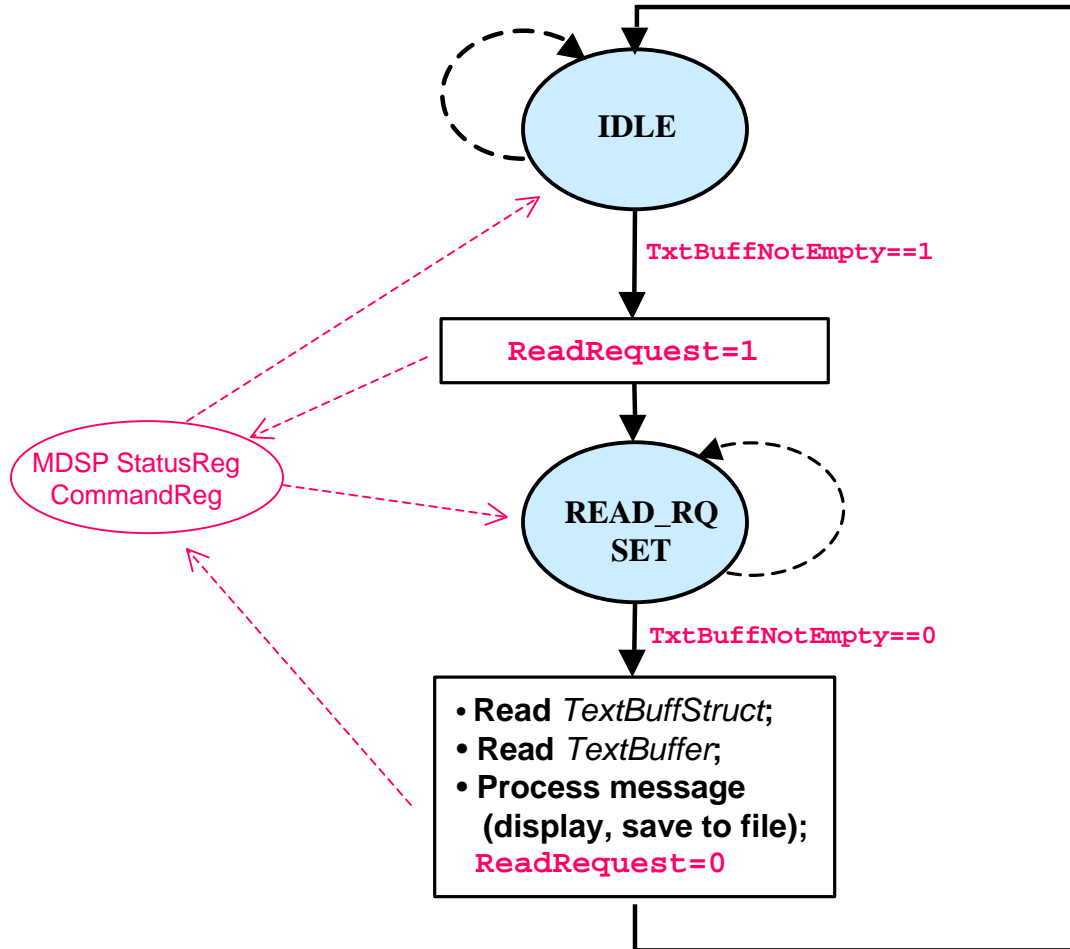


PrimListHandler[n] State Machine



TextBufferHandler[n] State Machine

- reads out and processes text messages from MasterDSP (error, info, diagnostic, xFer).



Thread priorities

- ***MainProcessThread(UserInterface) > the other threads***

- the user interface must be responsive at any moment!!

- ***CommunicationThread <= CmdListExecutonThread***

CommunicationLoopThread is still active -> the processor is always 100% busy. The *CmdListExecutonThreads* are active only for a limited period of time, so they can have a higher priority than *CommunicationLoop*.

To do

- complete and test the new “simpler” TestStand (next week),
- update for Pixels - i.e. implement pixel specific primitives, and work on TestStand integration into the pixel TurboDAQ system for initial ROD testing with pixel modules (September),
- prepare TestStand User Guide (September, October...).