

SATELLITE FLIGHT SOFTWARE TEAM

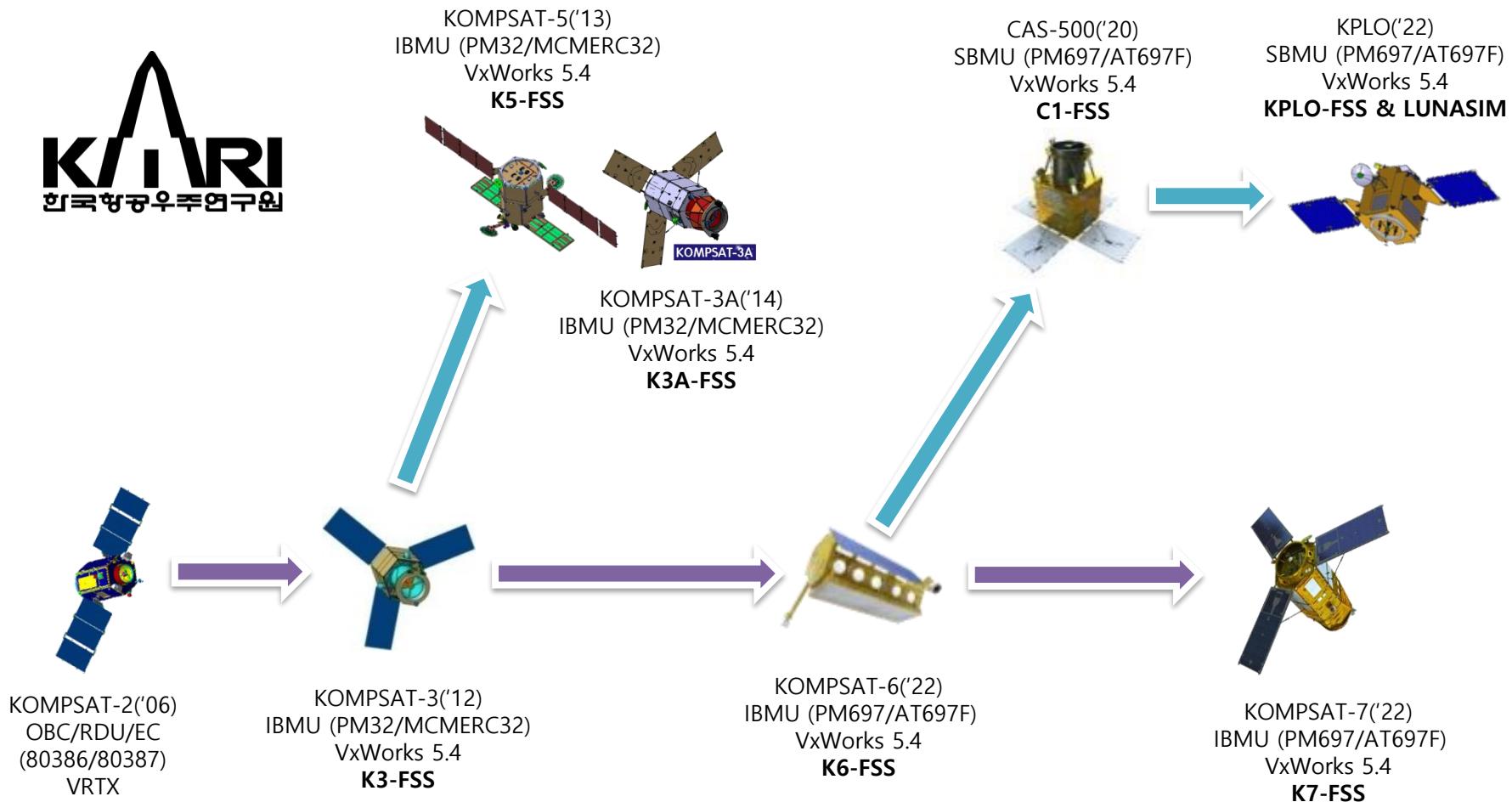
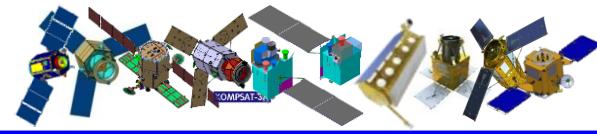


OVERALL KOMPSAT-FSS & LAYSIM

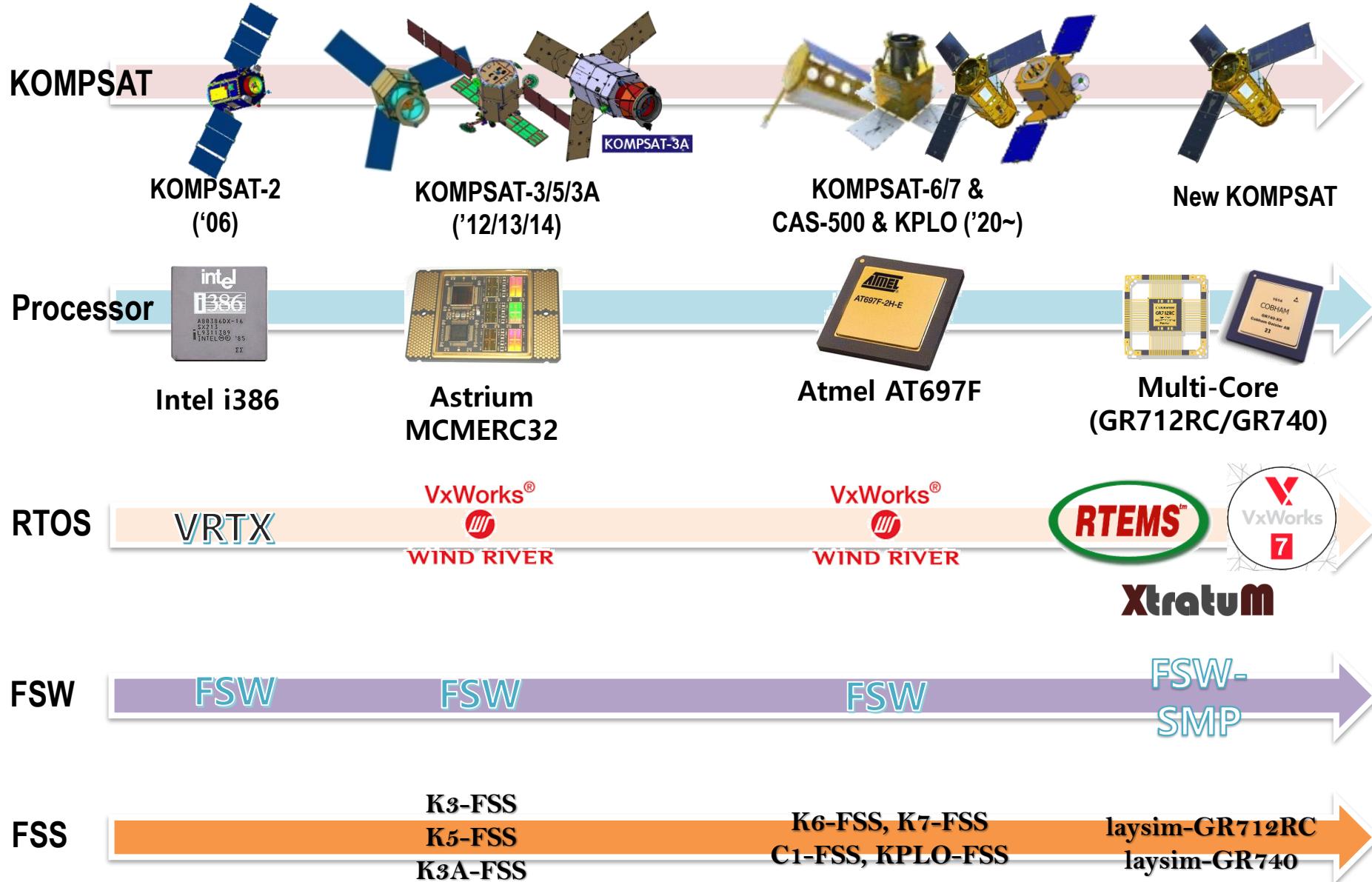
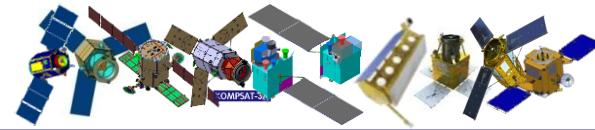
2020. 06. 18

Jong-Wook Choi
(jwchoi@kari.re.kr)

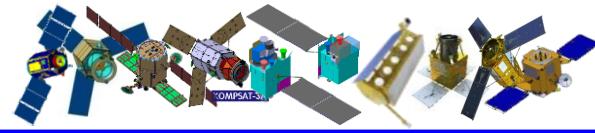
KOMPSAT Series



KOMPSATs Processor & FSW



laysim Processor Emulator



laysim-erc32

K3-FSS

K3A-FSS

K5-FSS

GK2A-FSS*

GK2B-FSS*

laysim-at697f

K6-FSS

K7-FSS

C1-FSS

KPLO-FSS

laysim-LEON3

laysim-LEON3MP

laysim-LEON3FT

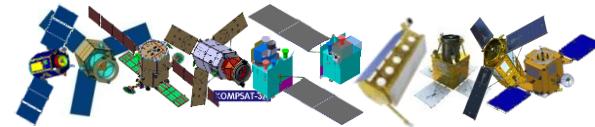
laysim-UT700

laysim-GR712RC

laysim-GR740

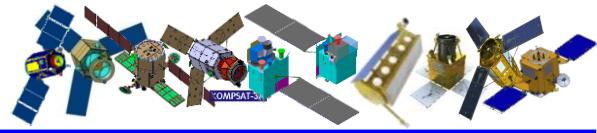
lay2cpu (E-FSS)

ERC32&LEON Processor Emulators

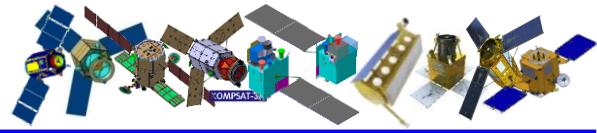


Emulator	Type	ERC32	LEON2	LEON3	LEON MP	Supplier	Remark
TSIM	Interpreter	○	○	○	X	COBHAM Gaisler	<ul style="list-style-type: none"> - Cycle True / Used for most ESA Projects - Not meet real-time performance for LEON2/3 (30MPIS) - Used for Initial KOMPSAT-3/5 FSS - GRSIM provides LEON MP Simulation - New TSIM3 beta versions are released
Leon-SVE	Interpreter	○	○	X	X	Spacebel	<ul style="list-style-type: none"> - Full representativity of LEON2-FT
SimERC32/SimLEON	Interpreter	○	○	○	X	Astrium/CNES	<ul style="list-style-type: none"> - Astrium Internal (SIMERC32 Emulator in SiMIX) - Gaia Real-Time Simulator
Sim-SCOC3	DBT	X	X	○	X	Astrium	<ul style="list-style-type: none"> - Spacecraft Controller On-a Chip with LEON3FT
leonSVF	H/W	X	○	○	X	Terma/Astrium	<ul style="list-style-type: none"> - 100% Representative - Realtime OBC Simulation at 60MHz+
ESOC Simulator	Interpreter	○	○	○	X	ESOC/VEGA	<ul style="list-style-type: none"> - Used for most ESOC/ESA ground system - New LEON2/LEON3 are supported
QERx	DBT	○	○	○	X	SciSys/FFQTECH	<ul style="list-style-type: none"> - Based on QEMU 0.9.1 - Galileo Constellation Operation Simulator
TEMU2	Interpreter	○	○	○	○	Terma	<ul style="list-style-type: none"> - New emulator based on LLVM framework - 200MIPS on a 3.5GHz PC - Support LEON4, PowerPC and ARM processor

Emulator	Type	ERC32	LEON2	LEON3	LEON3/4 MultiCore	Remark	
laysim	Interpreter	○	○	○	○ (GR740/GR712RC)	<ul style="list-style-type: none"> - Window & Linux Execution Platform - Source Level Debugging and Cycle True - Used for KOMPSAT Flight Software Simulator (FSS) - Expandable through user loadable modules (GR712RC, UT700) 	
laysim-DBT	DBT	○	○	○	○ (GR740/GR712RC)	<ul style="list-style-type: none"> - New DBT engine from scratch - High Performance and Cycle Accuracy (over 500MIPS @i7-6700K 4GHz) - Used for KOMPSAT FSS and Operation Simulator - Expandable through user loadable modules (GR712RC, UT700) 	
QEMU laysim	DBT	○	X	○	X	<ul style="list-style-type: none"> - Based on QEMU 1.0.1 - S/W development in VxWorks/RTEMS - Discontinued and outdated 	

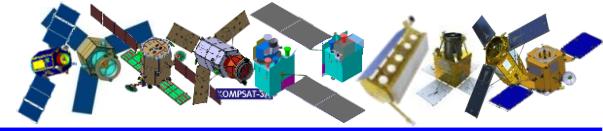


History of laysim Development

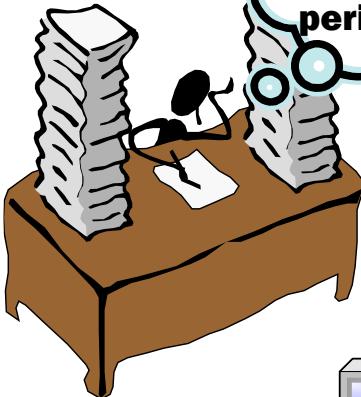


Problem ?

Anguish ?

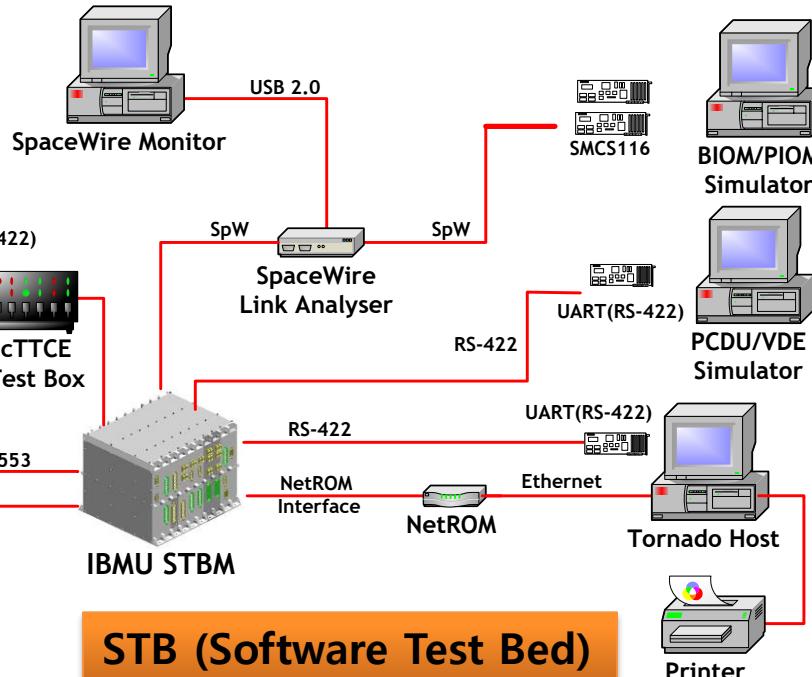


Limitation ?



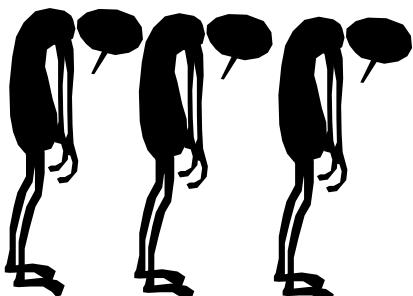
How to
shorten
period ?

How to debug
in real-time ?

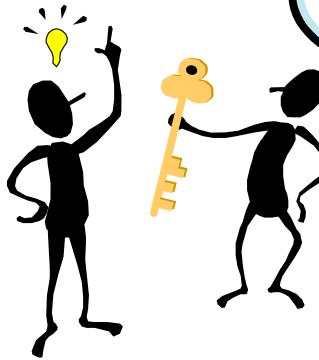
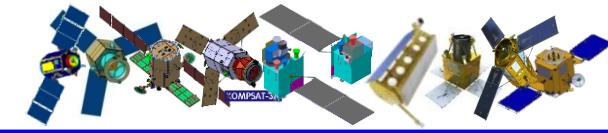


We want
to use
also!

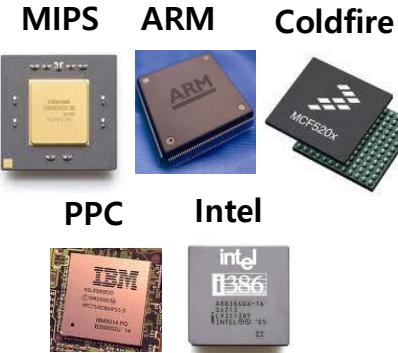
How to
cost down ?



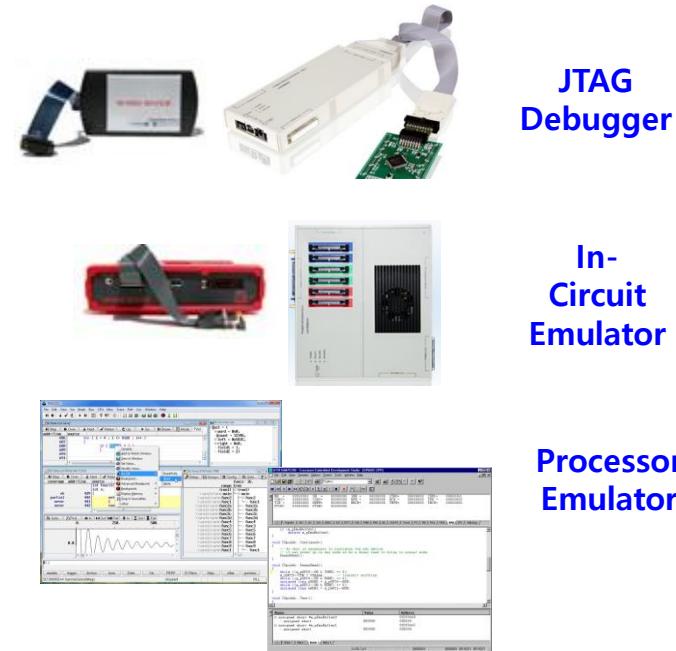
Solution ?



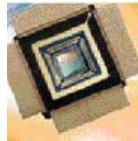
Use
Emulator
/Simulator



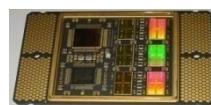
Rich Solution



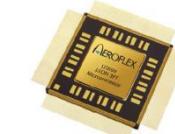
ERC32



MCMERC32



AT697F



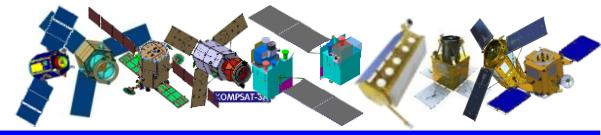
LEON3-FT

Poor Solution

No Working JTAG for MCM-ERC32

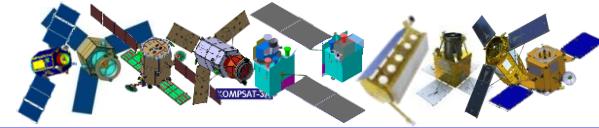
No In-Circuit Emulator

Only ERC32 & LEON2/3 Simulator
(TSIM) from Gaisler [Commercial]

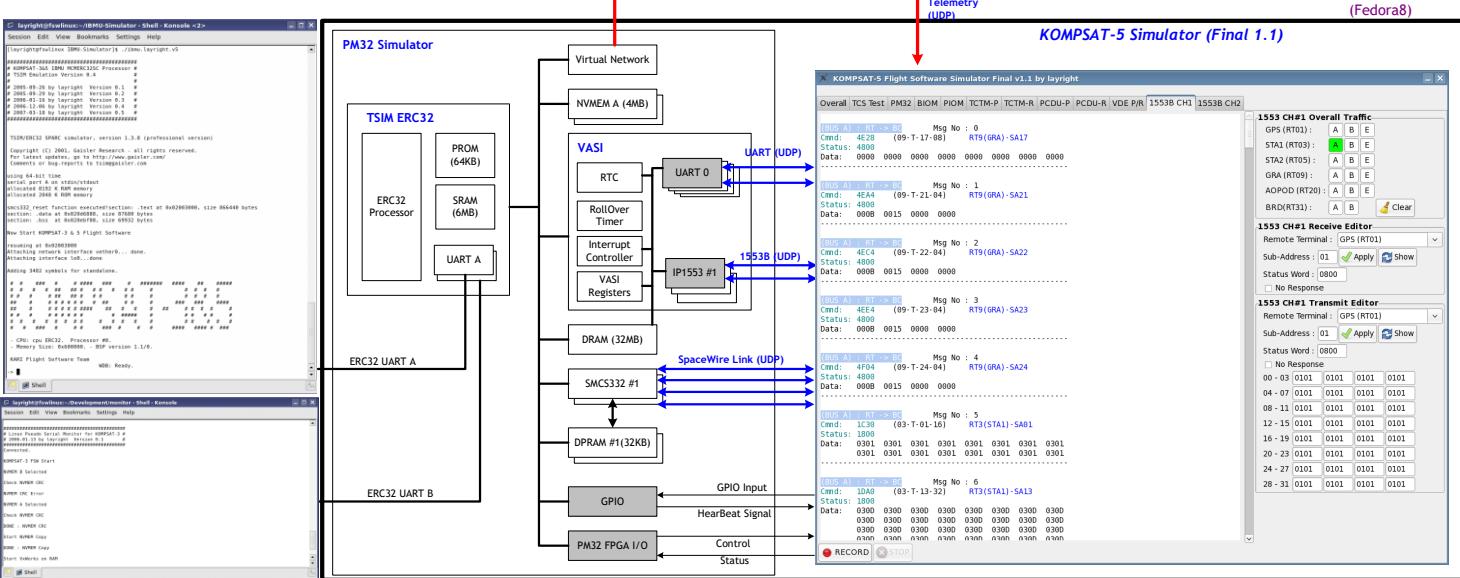
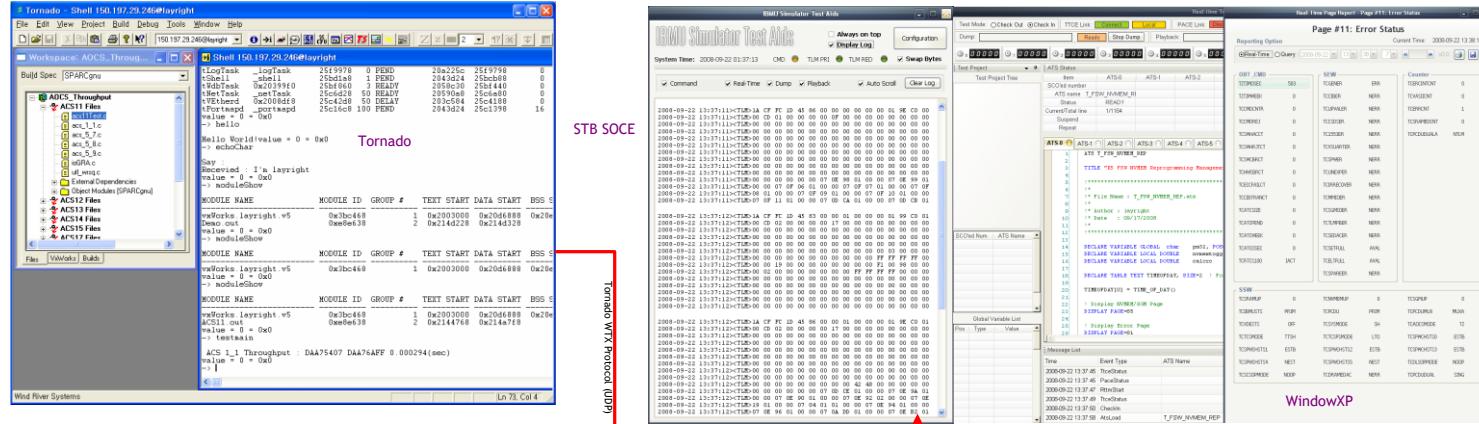


1st Solution & Limitation

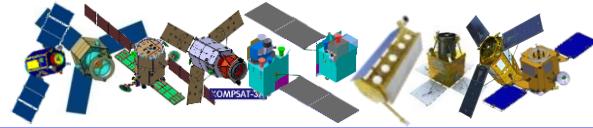
1st Solution [1/2]



- Construct KOMPSAT Flight Software Simulator with Commercial ERC32 Processor Emulator (TSIM-ERC32 from Gaisler)



1st Solution [2/2]



KOMPSAT-5 Flight Software Simulator Final v1.1 by layright

Overall TCS Test PM32 BIOM PIOM TCTM-P TCTM-R PCDU-P PCDU-R VDE P/R 1553B CH1 1553B CH2

(BUS A) : RT > BC Msg No : 0
Cmnd: 4E28 (09-T-17-08) RT9(GRA)-SA17
Status: 4800
Data: 0000 0000 0000 0000 0000 0000 0000 0000

(BUS A) : RT > BC Msg No : 1
Cmnd: 4EA4 (09-T-21-04) RT9(GRA)-SA21
Status: 4800
Data: 000B 0015 0000 0000

(BUS A) : RT > BC Msg No : 2
Cmnd: 4EC4 (09-T-22-04) RT9(GRA)-SA22
Status: 4800
Data: 000B 0015 0000 0000

(BUS A) : RT > BC Msg No : 3
Cmnd: 4EA4 (09-T-23-04) RT9(GRA)-SA23
Status: 4800
Data: 000B 0015 0000 0000

(BUS A) : RT > BC Msg No : 4
Cmnd: 4F04 (09-T-24-04) RT9(GRA)-SA24
Status: 4800
Data: 000B 0015 0000 0000

(BUS A) : RT > BC Msg No : 5
Cmnd: 1C30 (03-T-01-16) RT3(STA1)-SA01
Status: 1800
Data: 0301 0301 0301 0301 0301 0301 0301 0301
0301 0301 0301 0301 0301 0301 0301 0301

(BUS A) : RT > BC Msg No : 6
Cmnd: 1DA0 (03-T-13-32) RT3(STA1)-SA13
Status: 1800
Data: 0300 0300 0300 0300 0300 0300 0300 0300
0300 0300 0300 0300 0300 0300 0300 0300
0300 0300 0300 0300 0300 0300 0300 0300
0300 0300 0300 0300 0300 0300 0300 0300

```
layright@fswlinux:~/KOMPSAT-5-Simulator-Development/KOMPSAT-5-PM32-Simulat - Shell - Ko
```

Session Edit View Bookmarks Settings Help

[PM32 Simulator] VASI Reset
[PM32 Simulator] Virtual Network Reset

section: ../../obj/K3-prom.hex at 0x0, size 46392 bytes
section: ../../obj/K3-prom.hex at 0xf000, size 2472 bytes
entry point: 0x0
tsim> go
resuming at 0x00000000
0x24f83e0 (tMAIN): OBT:000007 - 0E98:INIT_GDA
0x24f83e0 (tMAIN): OBT:000007 - 0E99:INIT_TRAP
0x24f83e0 (tMAIN): OBT:000007 - 0F06:SPW CH#0 Established
0x24f83e0 (tMAIN): OBT:000007 - 0F07:SPW CH#1 Established
0x24f83e0 (tMAIN): OBT:000007 - 0F08:SPW CH#2 Established
0x24f83e0 (tMAIN): OBT:000007 - 0F09:SPW CH#3 Established
0x24f83e0 (tMAIN): OBT:000007 - 0F10:SPW CH#4 Link Error
0x24f83e0 (tMAIN): OBT:000007 - 0F11:SPW CH#5 Link Error
0x24f83e0 (tMAIN): OBT:000007 - 0116:SGM_P scrub disable before
0x24f83e0 (tMAIN): OBT:000007 - 0116:SGM_R scrub disable before
0x24f83e0 (): OBT:000007 - 0DCB:mmd init codea not found
0x24f83e0 (): OBT:000007 - 0DCB:mm p selected
0x24f83e0 (): OBT:000007 - 0DCE:mmd init completed
0x24f83e0 (): OBT:000007 - 0E9A:INIT_UART
0x24f71d0 (tBACK): OBT:000007 - 0E90:tSCP_Spawned
0x24f71d0 (tBACK): OBT:000007 - 0E92:ti1553_Spawned
0x24f71d0 (tBACK): OBT:000007 - 0E92:ti1553_Spawned
0x24ff250 (TSCP): OBT:000007 - 041E:RTCS_EXECUTE
0x24f71d0 (tBACK): OBT:000007 - 0E94:tONE_Spawned
0x24f71d0 (tBACK): OBT:000007 - 0E96:tTWO_Spawned
0x24f71d0 (tBACK): OBT:000007 - 0ADD:SRAM_EDAC_ENABLE
0x24f71d0 (tBACK): OBT:000007 - 0EB2:INIT_SGM_PATCH_TABLE
0x24f71d0 (tBACK): OBT:000007 - 0AD1:VAST RTC ENABLE
0x24f71d0 (tBACK): OBT:000007 - 0AE2:Enable PIT ISR
0x24f71d0 (tBACK): OBT:000007 - 0AE4:VAST 16Hz subminor tick e
0x24f71d0 (tBACK): OBT:000007 - 0AE7:Enable N slot down counte
0x24ff250 (TSCP): OBT:000008 - 081E:RTC_EXECUTION
0x24ff250 (TSCP): OBT:000058 - 0A04:No operation
0x24ff250 (TSCP): OBT:000058 - 0A04:No operation
0x24ff250 (TSCP): OBT:00005F - 081E:RTC_EXECUTION
0x24ff250 (TSCP): OBT:00005F - 0375:
0x24e4ba0 (tONE): OBT:000118 - 00A4:No operation

KOMPSAT-5 Flight Software Simulator Final v1.1 by layright

Overall TCS Test PM32 BIOM PIOM TCTM-P TCTM-R PCDU-P PCDU-R VDE P/R 1553B CH1 1553B CH2

ER32 GPIO (0x01FB800AC)

Bit 0 : Launch Abort Signal	0	(bit)
Bit 1 : Heart Beat Signal	0	(bit)
Bit 4 : Test/Flight Mode #1	<input checked="" type="radio"/> Test	<input type="radio"/> Flight
Bit 5 : EGSE Reprogramming	<input type="radio"/> Enable	<input checked="" type="radio"/> Disable
Bit 6 : PM32 Identification	1	

NVMEM Control Register (0x13000014)

Bit 0 : NVMEM Bank A&B Power On/Off	1	
Bit 1 : NVMEM Rep. Enable Latch Clear	1	
Bit 2 : NVMEM FSW Wrtie Enable #1	1	
Bit 3 : NVMEM FSW Wrtie Enable #2	1	

NVMEM Status Register (0x13000018)

Bit 0 : NVMEM Power Status of Bank A	0	
Bit 1 : NVMEM Power Status of Bank B	0	
Bit 2 : NVMEM Bank Selection from TCTM P	0	
Bit 3 : NVMEM Bank Selection from TCTM R	0	
Bit 4 : NVMEM En. SPC CMD Status from Tp	1	
Bit 5 : NVMEM En. SPC CMD Status from Tr	1	
Bit 6 : NVMEM En. SPC CMD Latch Status	1	
Bit 7 : NVMEM Write Enable Status	1	

GPS 1PPS Interval Counter (0x1300001C)

00000000	<input checked="" type="checkbox"/> Apply
----------	---

GPS 1PPS Tick Counter (0x13000020)

00000000	<input checked="" type="checkbox"/> Apply
----------	---

GPS 1PPS Status Register (0x13000024)

00000000	<input checked="" type="checkbox"/> Apply
----------	---

GPS 1PPS Delta-T Counter (0x13000060)

00000000	
----------	--

External RTC N Counter (0x13000028)

0026259F	
----------	--

External RTC Control Reg (0x1300002C)

Bit 0 : Sync Enable/Disable	0	
Bit 1 : Sync IRQ Select	0	
Bit 2 : Sync IRQ Enable/Disable	0	
Bit 3 : N Slot Divider Enable/Disable	1	

NVMEM Block Erase Start/End (0x1300003C)

00000000	
----------	--

NVMEM Page Read Start/End (0x13000040)

00000000	
----------	--

NVMEM Page Write Start/End (0x13000044)

00000000	
----------	--

NVMEM Write Protection/Spare (0x13000048)

00000002	
----------	--

SpaceWire Reset Register (0x13000010)

Bit 0 : SMCS116 Pri. Reset for both TCTMs	0	
Bit 1 : SMCS116 Red. Reset for both TCTMs	1	
Bit 2 : SMCS116 Reset for B-IOM	0	
Bit 3 : SMCS116 Reset for P-IOM	0	
Bit 4 : SMCS332 Reset	1	

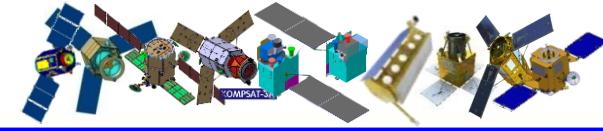
NVMEM Busy Status Register (0x13000058)

00000003	
----------	--

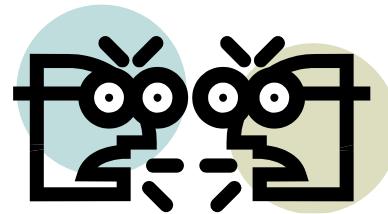
UART0 MUX Register (0x1300005C)

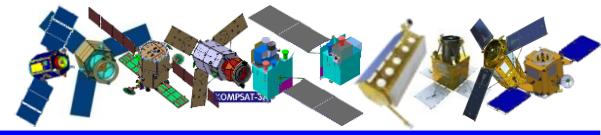
00000000	
----------	--

Limitation of 1st Solution



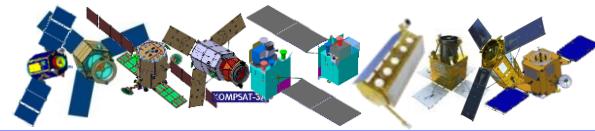
- **KOMPSAT Flight Software Simulator based on TSIM-ERC32/TSIM-LEON3**
 - Cost Problem : over 20,000 € (all FSW member should buy TSIM ? Another segments ?)
 - Real-time register & variable monitoring limitation
 - How can enhance or add test/debug interface into TSIM core ?
 - Difficulty for integration/develop of Satellite Simulator with another pre-developed avionics simulator
 - **In case of LEON2-FT processor, TSIM-LEON2 can not represent actual AT697F processor**
 - **In case of LEON3-FT processor, TSIM-LEON3 can not represent actual LEON3-FT processor**



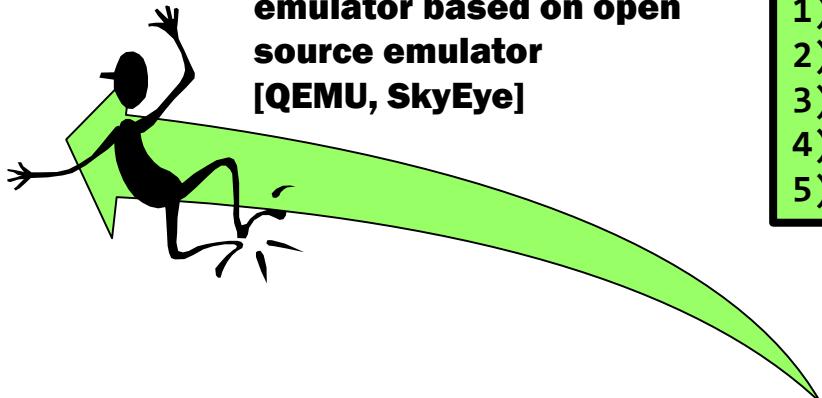


2nd Solution & Limitation

How to ?



**Develop ERC32&LEON2/3
emulator based on open
source emulator
[QEMU, SkyEye]**



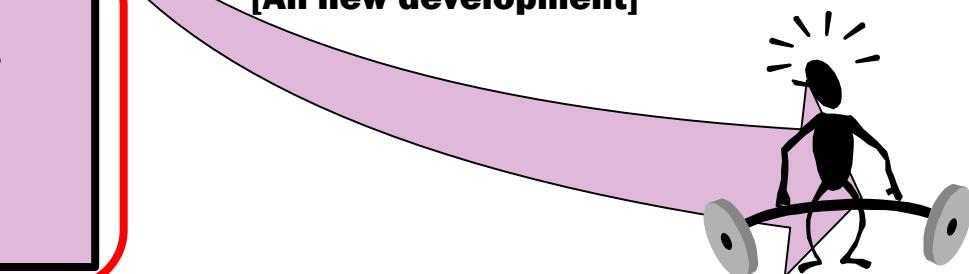
- 1) relatively easy
- 2) use IU/FPU core from open source emulator
- 3) implement only ERC32&LEON2/3 own functions
- 4) timing problem
- 5) interpretation or dynamic binary translation ?



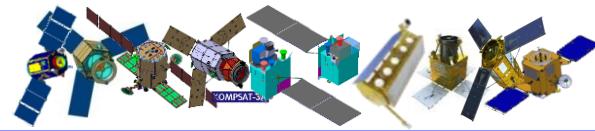
Escape from
TSIM

- 1) really hard
- 2) develop all for ERC32&LEON2/3 cores
- 3) verification problem
- 4) add everything if want
- 5) cycle accuracy
- 6) Interpretation or DBT ?

**Develop new ERC32&LEON2/3 emulator
[All new development]**



2nd Solution

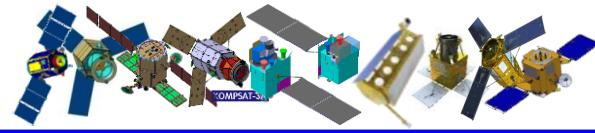


- Develop ERC32 & LEON2/3 Processor Emulator based on **Interpreter** method
 - **laysim-erc32 & laysim-leon3** (based on **interpretation method**)
 - TSIM also uses Interpretation method
- **Interpreter Execution Flow**
 1. Fetch **one guest instruction** from guest memory image
 2. Decode and dispatch to corresponding emulation unit
 3. Execute the functionality of that instruction and modify some related system states, such as simulated register values
 4. Increase the guest PC (Program Counter Register) and then repeat this process again
- **Pros & Cons**
 - Pros : Easy to implement
 - Cons : **Poor performance / Real-Time Performance Problem at high CPU clock**

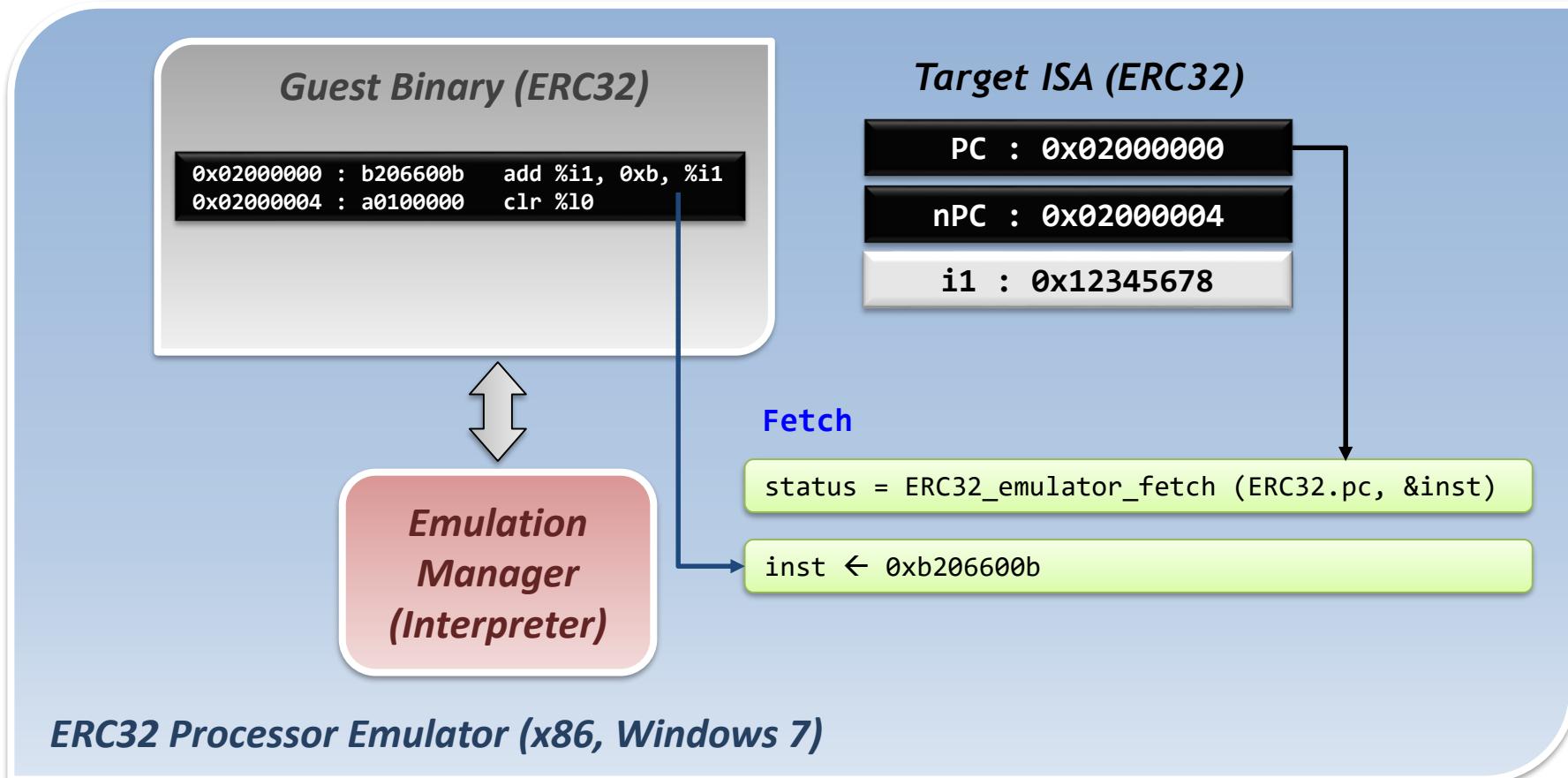
➔ ERC32 processor runs at 20MHz (Max 12DMIPS)

➔ **ERC32 processor emulator based on Interpreter can meet real-time performance because of low CPU clock**

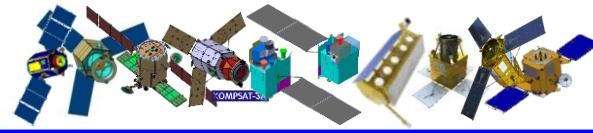
Interpreter [1/3]



- Fetch one instruction from guest binary (ERC32)
 - Get one instruction pointed by PC (Program Counter)



Interpreter [2/3]



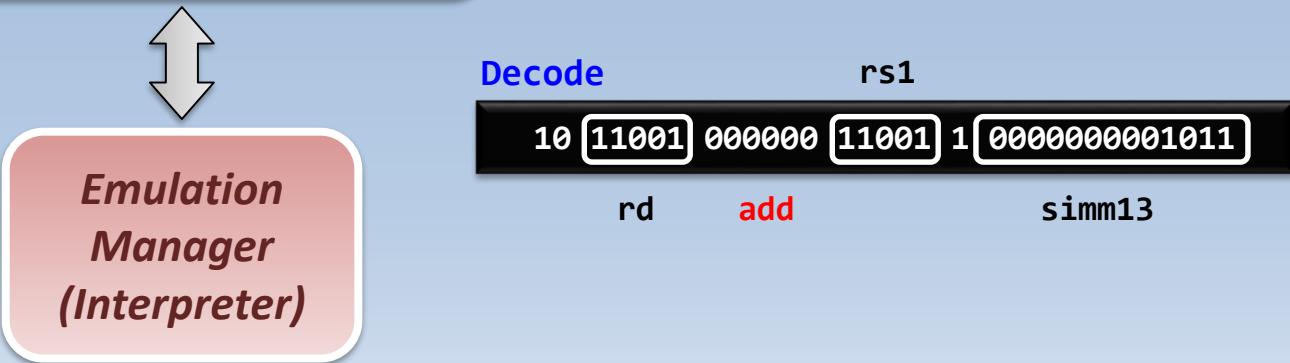
- **Decode fetched instruction**
 - Is it ADD, AND, XOR or MOV ?

Guest Binary (ERC32)

```
0x02000000 : b206600b    add %i1, 0xb, %i1
0x02000004 : a0100000    clr %10
```

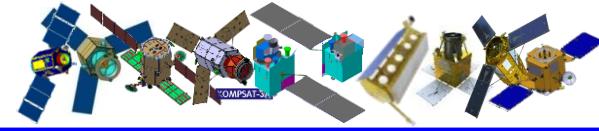
Target ISA (ERC32)

PC : 0x02000000
nPC : 0x02000004
i1 : 0x12345678

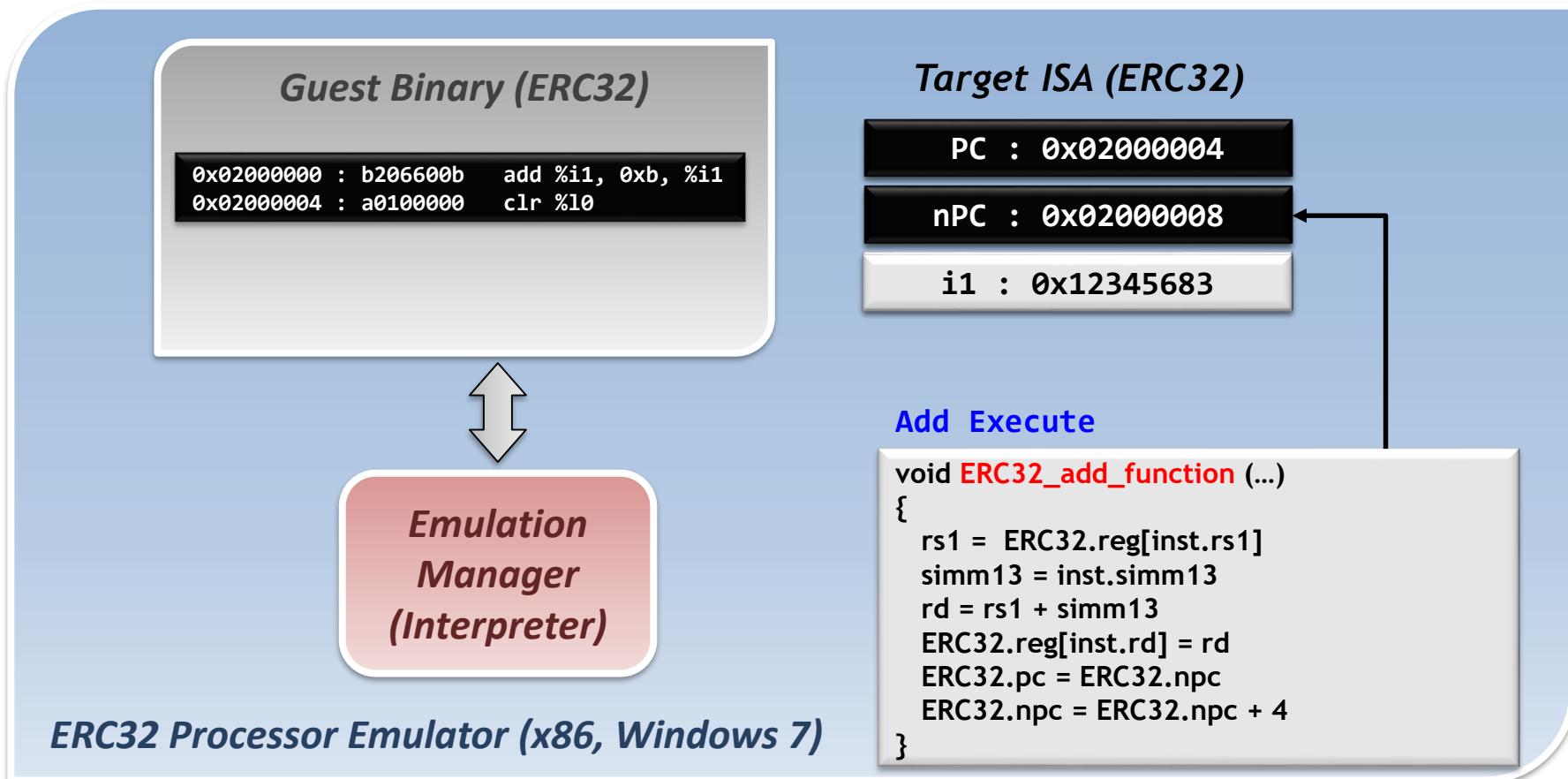


ERC32 Processor Emulator (x86, Windows 7)

Interpreter [3/3]

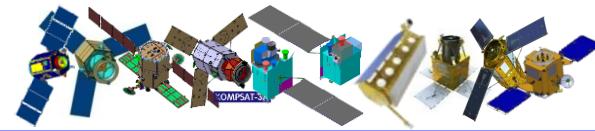


- Execute Emulation Function
 - Call corresponding C function



LEON3-FT Processor Emulator

[laysim-LEON3 (based on Interpreter)]



GUI based LEON3-FT Processor Emulator : laysim-leon3 v0.01 by layright [SCU-DM model]

IU FPU Cycles : 00000000000021226426

File Edit View Debug Tools Window Help

Load File : test01.exe

IU Reg. FPU Reg. GRLIB

Register Value

Register	Value																																				
PC	0x40001260																																				
nPC	0x40001264																																				
PSR	0xF34010E6																																				
WIM	0x00000001																																				
TBR	0x40000050																																				
Y	0x00000000																																				
Global																																					
Win 0																																					
Win 1																																					
Win 2																																					
Win 3	Register View																																				
Win 4																																					
Win 5																																					
Win 6	<table border="1"><tr><td>o0</td><td>0x00000007</td></tr><tr><td>o1</td><td>0x00000001</td></tr><tr><td>o2</td><td>0x00000002</td></tr><tr><td>o3</td><td>0x00000003</td></tr><tr><td>o4</td><td>0x4000ACAO</td></tr><tr><td>o5</td><td>0x00084D60</td></tr><tr><td>o6/sp</td><td>0x403FFD98</td></tr><tr><td>o7</td><td>0x400011F8</td></tr><tr><td>l0</td><td>0x40009DA8</td></tr><tr><td>l1</td><td>0x4000A800</td></tr><tr><td>l2</td><td>0x00000000</td></tr><tr><td>l3</td><td>0x00000000</td></tr><tr><td>l4</td><td>0x00000000</td></tr><tr><td>l5</td><td>0x00000000</td></tr><tr><td>l6</td><td>0x00000000</td></tr><tr><td>l7</td><td>0x00000000</td></tr><tr><td>l8</td><td>0x00000000</td></tr><tr><td>l9</td><td>0x40009908</td></tr></table>	o0	0x00000007	o1	0x00000001	o2	0x00000002	o3	0x00000003	o4	0x4000ACAO	o5	0x00084D60	o6/sp	0x403FFD98	o7	0x400011F8	l0	0x40009DA8	l1	0x4000A800	l2	0x00000000	l3	0x00000000	l4	0x00000000	l5	0x00000000	l6	0x00000000	l7	0x00000000	l8	0x00000000	l9	0x40009908
o0	0x00000007																																				
o1	0x00000001																																				
o2	0x00000002																																				
o3	0x00000003																																				
o4	0x4000ACAO																																				
o5	0x00084D60																																				
o6/sp	0x403FFD98																																				
o7	0x400011F8																																				
l0	0x40009DA8																																				
l1	0x4000A800																																				
l2	0x00000000																																				
l3	0x00000000																																				
l4	0x00000000																																				
l5	0x00000000																																				
l6	0x00000000																																				
l7	0x00000000																																				
l8	0x00000000																																				
l9	0x40009908																																				

C Source Level Debugging

```
while () {  
    lay1++;  
    40001200 0310002b sethi %hi(0x4000ac00)  
    40001204 9a1060a4 or %g1, 0xa4, %0  
    40001208 0310002b sethi %hi(0x4000ac00)  
    4000120c 821060a4 or %g1, 0xa4, %0  
    40001210 c2004000 ld [%g1], %g1  
    40001214 82006001 add %g1, 0x1, %g0  
    40001218 c2234000 st %g1, [%o5]  
  
    lay2++;  
    4000121c 0310002b sethi %hi(0x4000ac00)  
    40001220 981060a0 or %g1, 0xa0, %0  
    40001224 0310002b sethi %hi(0x4000ac00)  
    40001228 9a1060a4 or %g1, 0xa4, %0  
    4000122c 0310002b sethi %hi(0x4000ac00)  
    40001230 821060a8 or %g1, 0xa8, %0  
    40001234 da034000 ld [%o5], %o5  
    lay3 = lay1;  
    40001238 0310002b sethi %hi(0x4000ac00)  
    4000123c 981060a0 or %g1, 0xa0, %0  
    40001240 0310002b sethi %hi(0x4000ac00)  
    40001244 9a1060a4 or %g1, 0xa4, %0  
    40001248 0310002b sethi %hi(0x4000ac00)  
    4000124c 821060a8 or %g1, 0xa8, %0  
    40001250 da034000 ld [%o5], %o5  
    40001254 c2004000 ld [%g1], %g1  
    40001258 82034001 add %o5, %g1, %g0  
    > 4000125c c2230000 st %g1, [%o4]  
    40001260 10bf0ffe8 ba 0x40001200  
    40001264 01000000 nop
```

Toggle Breakpoint
Show Breakpoints
Remove All Breakpoints
Set PC Here
Add Watch
Run to Cursor

Breakpoint Window

Num.	Address	Name
0	40001170	Breakpoint

Watch Window

Num.	Address	Value	Name
0	4000aca4	00084d60	lay1
1	4000aca8	00084d61	lay2
2	4000aca0	00109ac1	lay3

Symbol Table Window

Num.	Type	Address	Size	Name
187	bss	4000a97c	00000080	inirq
188	bss	4000a9fc	00000004	no_inirq_check
189	bss	4000aa00	00000200	_irqtbl
190	bss	4000ac00	00000080	_irqtbl
191	bss	4000ac80	00000004	_lbst_pthread_mutexattr
192	bss	4000ac84	00000004	_lbst_pthread_mutexattr
193	bss	4000ac88	00000004	_lbst_pthread_mutexattr
194	bss	4000ac8c	00000004	_lbst_pthread_mutex_unl
195	bss	4000ac90	00000004	_lbst_pthread_mutex_loc
196	bss	4000ac94	00000004	_lbst_pthread_mutex_tryl
197	bss	4000ac98	00000004	_lbst_pthread_mutex_des
198	bss	4000ac9c	00000004	_lbst_pthread_mutex_init
199	bss	4000aca0	00000004	lay3
200	bss	4000aca4	00000004	lay1
201	bss	4000aca8	00000004	lay2
202	bss	4000acac	00000004	errno
203	bss	4000acb0	00000004	heap_ptr
204	bss	4000acc0	00000008	xtime

Symbol Table

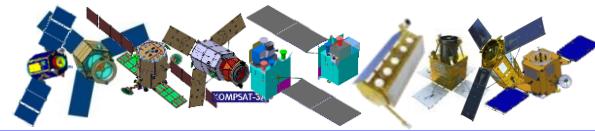
Console UART 1

```
(laysim) stop  
(laysim) t  
21226423 4000125c c2230000 st %g1, [%o4]  
  
INS LOCALS OUTS GLOBALS  
0: 00000000 40009D8 00000007 00000000  
1: 40009D08 4000A800 00000001 00109AC1  
2: 00000000 00000000 00000002 4000A8A0  
3: 00000000 00000000 00000003 FFFFFFF8  
4: 00000000 00000000 4000ACAO 40009000  
5: 00000000 00000000 00084D60 4000A874  
6: 403FFE00 00000000 403FFD98 4000937C  
7: 40001064 00000000 400011F8 00000000  
  
psr: F34010E6 [n:0 z:1 v:0 c:0 ef:1 pil:0 s:1 ps:1 et:1 cu:  
wim: 00000001 tbr: 40000050 y: 00000000  
  
pc: 40001260 10bf0ffe8 ba 0x40001200  
npc: 40001264 01000000 nop
```

Control Console

Construct KOMPSAT FSS

[base on laysim-ERC32]



KOMPSAT-3 Flight Software Simulator v1.2 by layright

UART-A UART-B PM32 1553B

```

K3-FSS - binary file size : 63914 bytes
2013-10-25 08:33:42 K3-FSS K3 NVMem File is loaded - C:\MinGW\msys\1.0\home\jwchoi\K3_Flight_Software_Simulator\example\exam...
2013-10-25 08:33:42 K3-FSS - Calculated NVMem CRC32 : 0x59C89FCF
2013-10-25 08:33:42 K3-FSS - Loaded NVMem length : 0x001ED000
2013-10-25 08:33:42 K3-FSS No K3 RAM File
2013-10-25 08:33:42 K3-FSS K3 Config File is loaded - C:\MinGW\msys\1.0\home\jwchoi\K3_Flight_Software_Simulator\exam...
2013-10-25 08:33:42 K3-FSS ERC32 ROM File is loaded, starts from ROM
2013-10-25 08:33:44 K3-FSS K3-FSS is running
2013-10-25 08:33:44 K3-FSS RU is Active! After 3mins RU will monitor H/B failure
2013-10-25 08:33:46 K3-FSS VASIM/PM32 is reset
2013-10-25 08:34:04 K3-FSS RU starts H/B Monitoring
    
```

KOMPSAT-3 Overall Status

PM32	P	R
TCTM Primary	ON	OFF
TCTM Redundant	ON	OFF
BIOM	P	R
PIOM	P	R
PCDU Primary	ON	OFF
PCDU Redundant	ON	OFF
VDE Primary	ON	OFF
VDE Redundant	ON	OFF
GRA P.S #1	ON	OFF
GRA P.S #2	ON	OFF
Star Tracker 1	ON	OFF
Star Tracker 2	ON	OFF
GPS Receiver 1	ON	OFF
GPS Receiver 2	ON	OFF
CEU Primary	ON	OFF

PM32/IBMU Register Window

Address	Name	Value	Description
0x01F800AC	ERC32 GPIO	00000070	Read/Write
0x13000000	PM32 Direct Output	000000FF	Write Only
0x13000004	PM32 Direct Input	00000001	Read Only
0x13000008	PIT Control Register	00000000	Write Only
0x1300000C	PIT Status Register	00000000	Read Only
0x13000010	SpaceWire Reset Register	00000012	Write Only
0x13000014	NVMEM Control Register	0000000F	Write Only
0x13000018	NVMEM Status Register	000000F0	Read Only
0x1300001C	GPS 1PPS Interval Counter	00000000	Read Only
0x13000020	GPS 1PPS Tick Counter	00000000	Read Only
0x13000024	GPS 1PPS Status Counter	00000000	Read Only
0x13000028	External RTC N Counter	0026259F	Write Only
0x1300002C	External RTC Control Register	00000008	Write Only
0x13000030	External SW Data Register	00000001	Write Only
0x1300003C	NVMEM Block Erase Start/End Flag	00000000	Write Only
0x13000040	NVMEM Page Read Start/End Flag	00000000	Write Only
0x13000044	NVMEM Page Write Start/End Flag	00000000	Write Only
0x13000048	NVMEM Write Protection Register	00000002	Write Only
0x13000058	NVMEM Busy Status Register	00000003	Read Only
0x1300005C	UART0 MUX Register	00000000	Write Only
0x13000060	GPS 1PPS Delta-T Counter	00000000	Read Only

ERC32 GPIO (0x01F800AC)

- Bit 00 : Launch Abort Signal
- Bit 01 : Heart Beat Signal
- Bit 02 :
- Bit 03 :
- Bit 04 : Test/Flight Mode #1 (Test=0, Flight=1)
- Bit 05 : EGSE Reprogramming (Enable=0, Disable=1)
- Bit 06 : PM32 Identification (Red=0, Pri=1)
- Bit 07 :

KOMPSAT-3 1553B Monitor

1553 CH#1 Overall Traffic

GPS (RT01)	A	B	E
STA1 (RT03)	A	B	E
STA2 (RT05)	A	B	E
GRA (RT09)	A	B	E
CEU (RT21)	A	B	E
IDHU (RT22)	A	B	E
XAA1 (RT23)	A	B	E
XAA2 (RT24)	A	B	E

1553 CH#1 Receive Editor

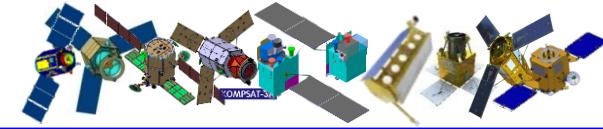
Remote Terminal : GPS (RT01)
Sub-Address : 01 Apply
Status Word : 0800
 No Response

1553 CH#1 Transmit Editor

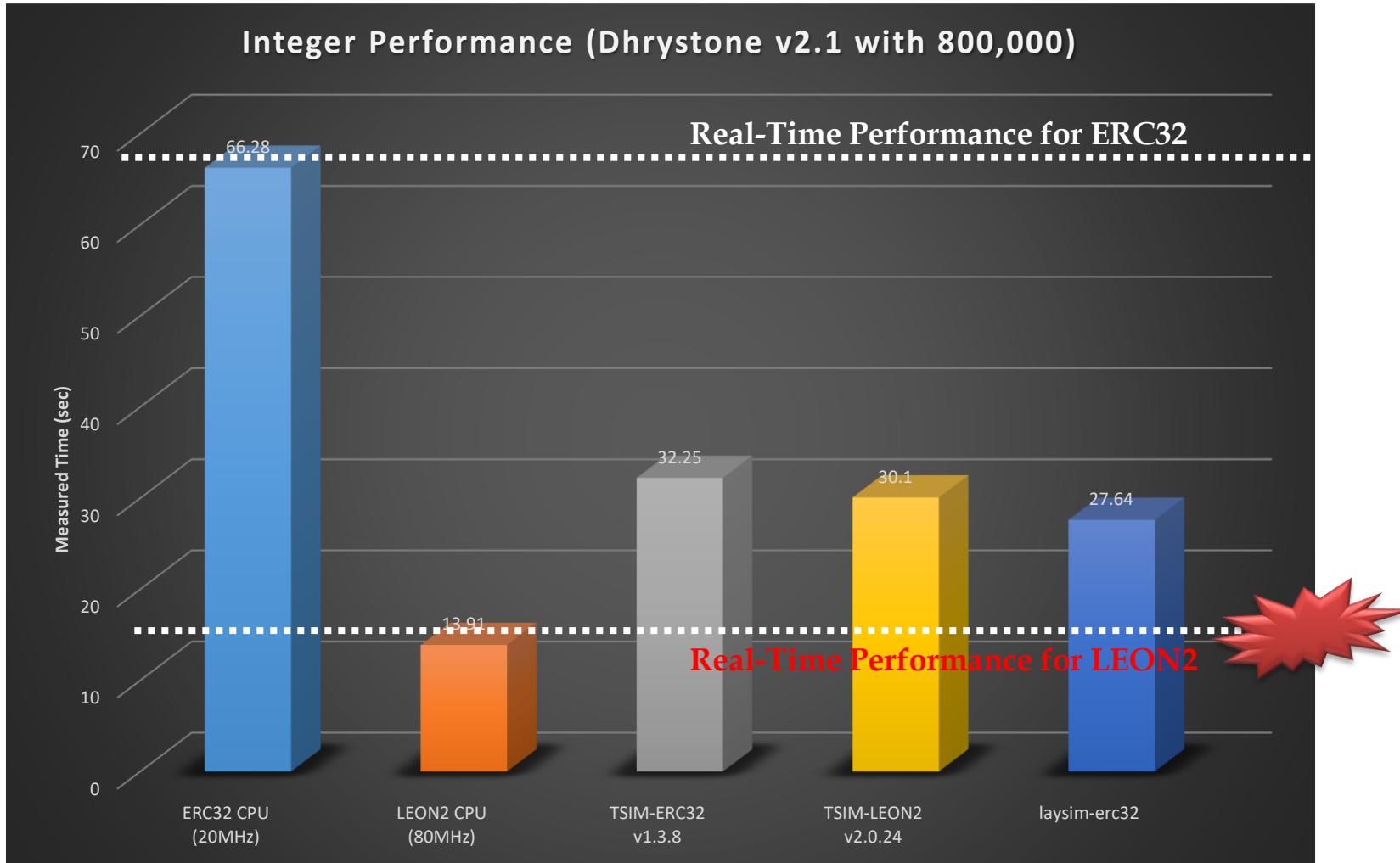
Remote Terminal : GPS (RT01)
Sub-Address : 01 Apply
Status Word : 0800
 No Response

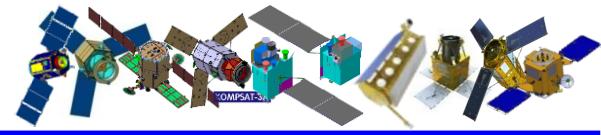
00 - 03	0000	0000	0000	0000
04 - 07	0000	0000	0000	0000
08 - 11	0000	0000	0000	0000
12 - 15	0000	0000	0000	0000
16 - 19	0000	0000	0000	0000

Limitation of 2nd Solution



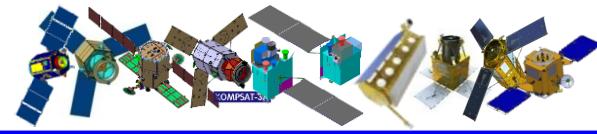
- **Real-Time Performance Problem** on new Processor
 - AT697F (LEON2-FT) / SCOC3 (LEON3-FT) (run @ 80~100MHz)



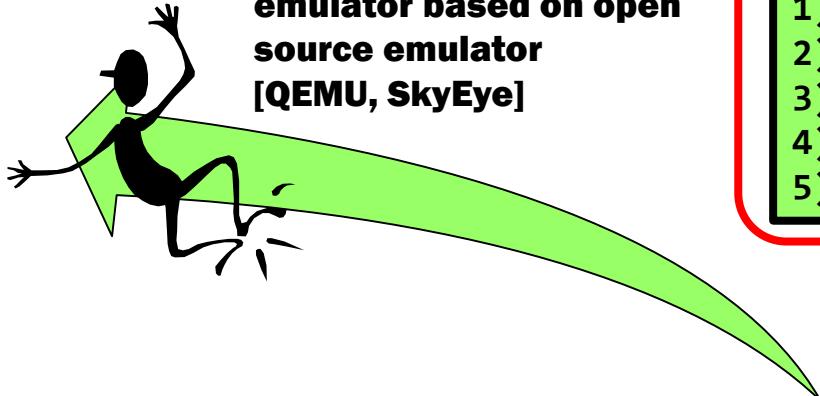


3rd Solution & Limitation

How to ?



**Develop ERC32&LEON2/3
emulator based on open
source emulator
[QEMU, SkyEye]**



- 1) relatively easy
- 2) use IU/FPU core from open source emulator
- 3) implement only ERC32&LEON2/3 own functions
- 4) timing problem
- 5) interpretation or dynamic binary translation ?

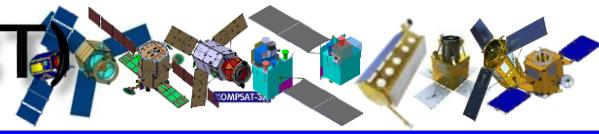


- 1) really hard
- 2) develop all for ERC32&LEON2/3 cores
- 3) verification problem
- 4) add everything if want
- 5) cycle accuracy
- 6) Interpretation or DBT ?

**Develop new ERC32&LEON2/3 emulator
[All new development]**



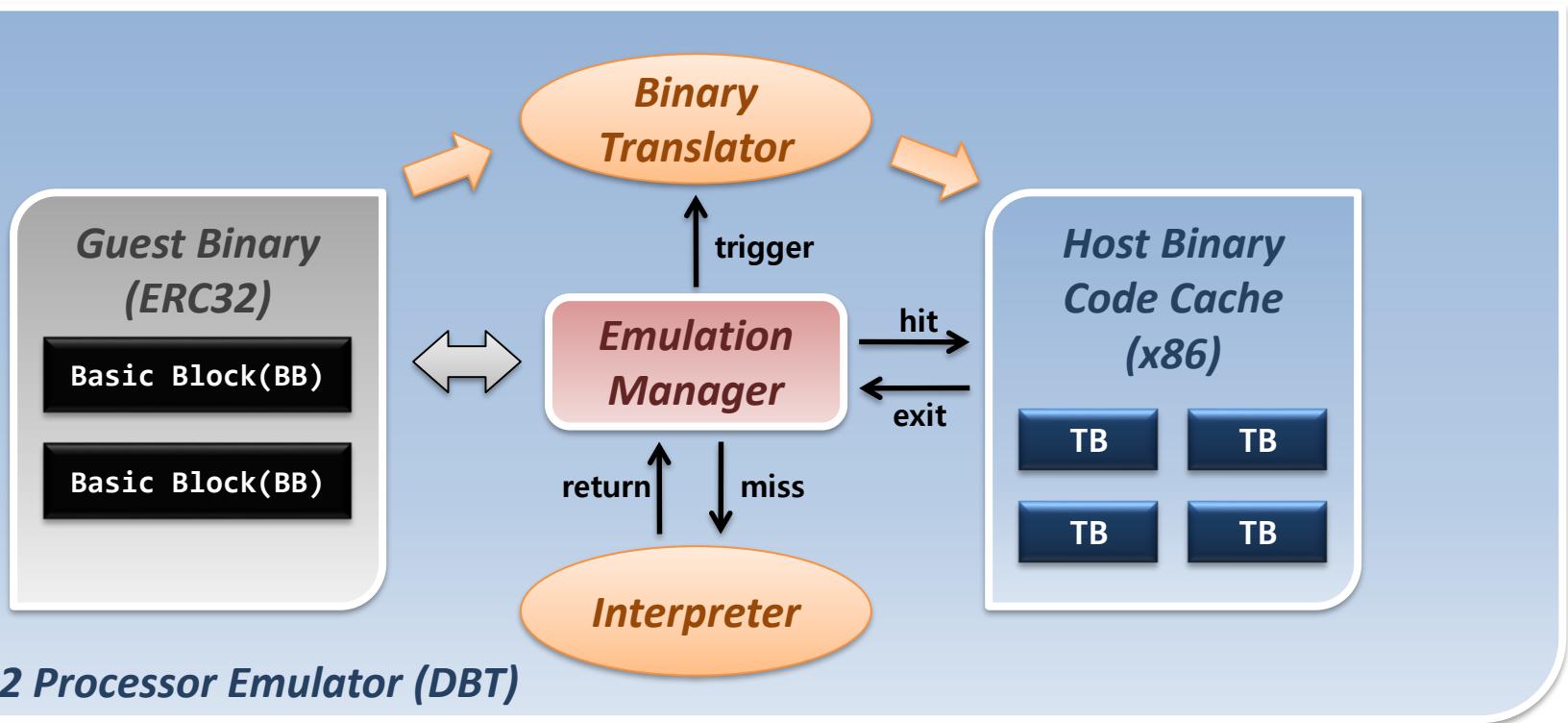
Dynamic Binary Translation (DBT)



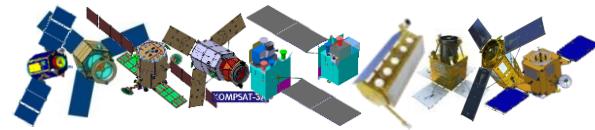
- Dynamic Binary Translation

1. First time execution, no translated code in code cache
2. No translated block (TB) matching, then directly interpret the guest (ERC32) instruction
3. As a basic block discovered, trigger the binary translation module
4. Translate guest (ERC32) basic block to host (x86) binary, and place it in the code cache
5. **Next time execution, run the translated block (TB) in the code cache**

➔ DBT is 5 ~ 10 times faster than interpretation, but instruction cycle & I/O timing issue



3rd Solution [1/2]



- Develop ERC32/LEON3 Processor Emulator based on **QEMU 1.0.1**
 - QEMU is generic and open source machine emulator based on DBT
 - **QEMU laysim-erc32 & QEMU laysim-leon3**

The screenshot shows the QEMU laysim interface running on a Windows host. The main window displays assembly code for a C program involving local variables and function pointers. Below the code is a memory dump showing the state of memory at address 0x0. The bottom left shows a task list with various processes like tExcTask, tLogTask, tShell, etc. The bottom right shows a timeline of tasks over time.

QEMU laysim (laysim-erc32)

```
dev-prop: UART-A = serial0
dev-prop: uartar (0x01F800E0) = 0x00000000
dev-prop: UART-B = serial1
dev-prop: uartrbr (0x01F800E4) = 0x00000000
dev-prop: uartsr (0x01F800E8) = 0x00060006
irq 4
mmio 00000000000000000000000000000000
```

(qemu) info registers

INS	LOCALS	OUTS	GLOBALS
0: 0208E800	FFFFF800	0208FA50	00000000
1: 0208F400	02009ECC	000000A8	114000C4
2: 0208E800	02009EFO	000000A8	023BC0C4
3: 00000001	0208E800	02080C00	00000000

QEMU laysim (laysim-erc32)

```
XXXXXXXX J XXXXX J XXXXX J XXXXX J XXXXX J XXXXX J XXXXX J XXXXX
XXXXXXXX J XXXXX J XXXXX
XXXXXXXX J XXXXX J XXXXX
XXXXXXXX J XXXXX J XXXXX
XXXXXXXX J XXXXX J XXXXX
Development System
XXXXXXXX J XXXXX J XXXXX
XXXXXXXX J XXXXX J XXXXX
UxWorks version 5.4
XXXXXXXX J XXXXX J XXXXX
KERNEL: WIND version 2.5
XXXXXXXX J XXXXX J XXXXX
Copyright Wind River Systems, Inc., 1984-1999
CPU: lay-ERC32SC. Processor #0.
Memory Size: 0x400000. BSP version 1.2/0.3.
WDB: Ready.

-> i
```

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tExcTask	_excTask	23fc2f8	0	PEND	2065b04	23fc118	0	0
tLogTask	_logTask	23f9978	0	PEND	2065b04	23f9798	0	0
tShell	_shell	23b3928	1	READY	203344c	23b3308	0	0
tWdbTask	20207b8	23b5b48	3	PEND	2028804	23b57a8	0	0
tNetTask	_netTask	23bbba38	50	PEND	2028804	23bb860	0	0
tEtherd	2004bf8	23b9028	50	READY	2011504	23b8ed8	0	0

value = 0 = 0x0

->

Tornado – layright_pc.aerospace.kari.re.kr

File Edit View Project Build Debug Tools Window Help

192.168.0.2@layright_pc

Watch 1 Watch 2 Watch 3 Watch 4

Name Value

exe_local_ptr 0x0

var1 0 'W000'

var2 0

var3 0

function1 0

 *exe_local_p...{void ()} 0

function2 0

 *exe_local_p...{void ()} 0

function3 0

 *exe_local_p...{void ()} 0

EXE_CSC_TYPE exe_local_type;

layright_pc.aerospace.kari.re.kr

0.76 0.80 0.84 0.88 0.92

INT13

tExcTask

tLogTask

tShell

tWdbTask

tNetTask

tEtherd

tWvRBuffMgr

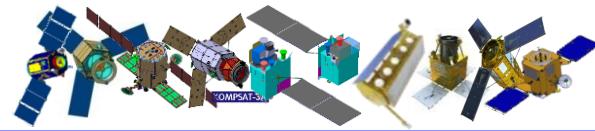
idle

INT13 : 0,949918 : intExitK ()

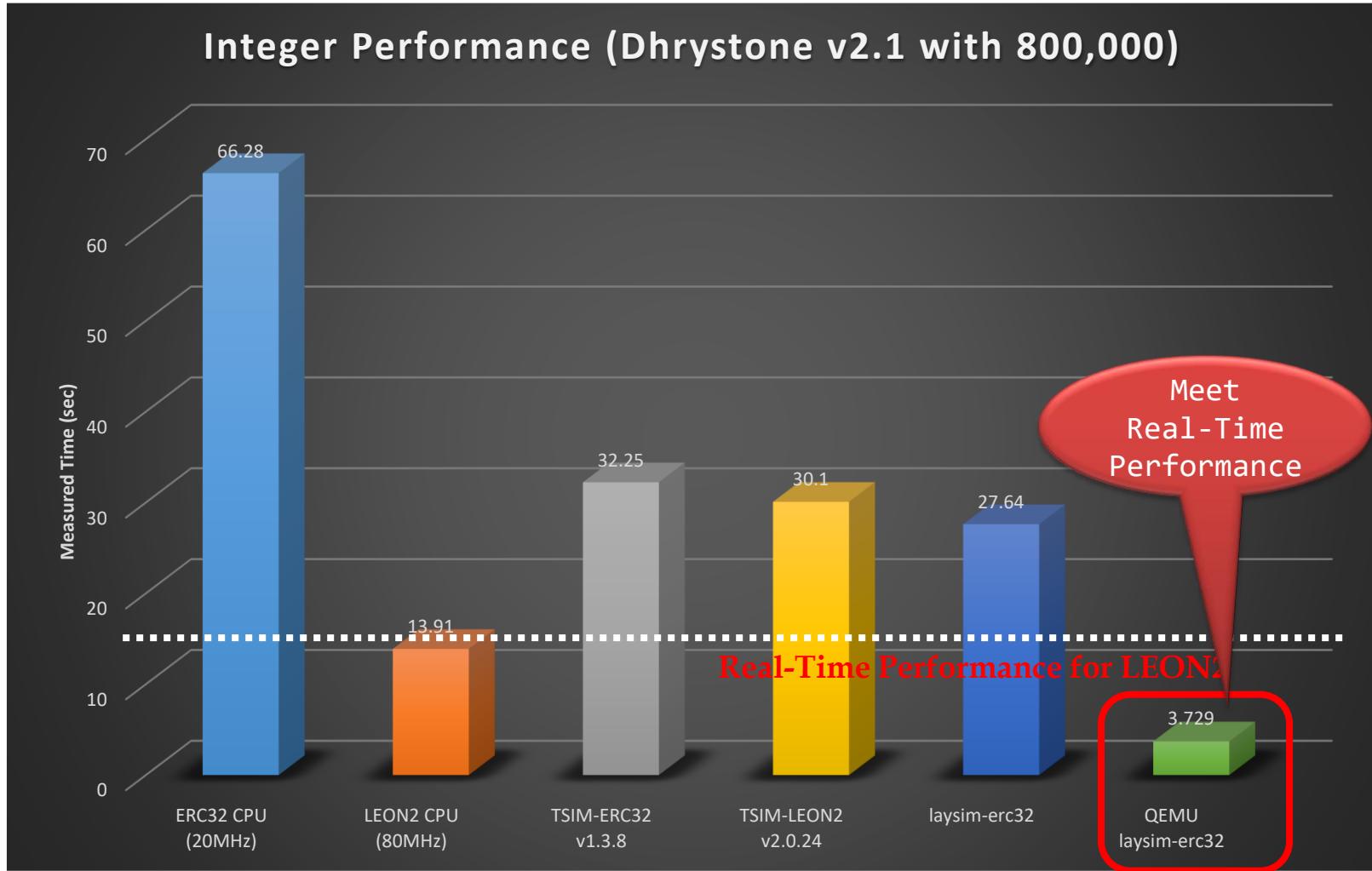
Wind River Systems

x6 (0x23a7128) 192.168.0.2@layright_pc : Stopped

3rd Solution [2/2]



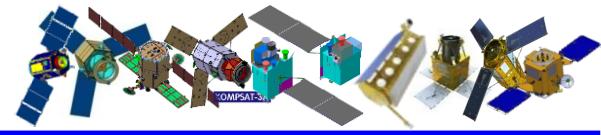
- QEMU laysim-ERC32/LEON3 show high-performance than interpretation emulator. **But**



Limitation of 3rd Solution

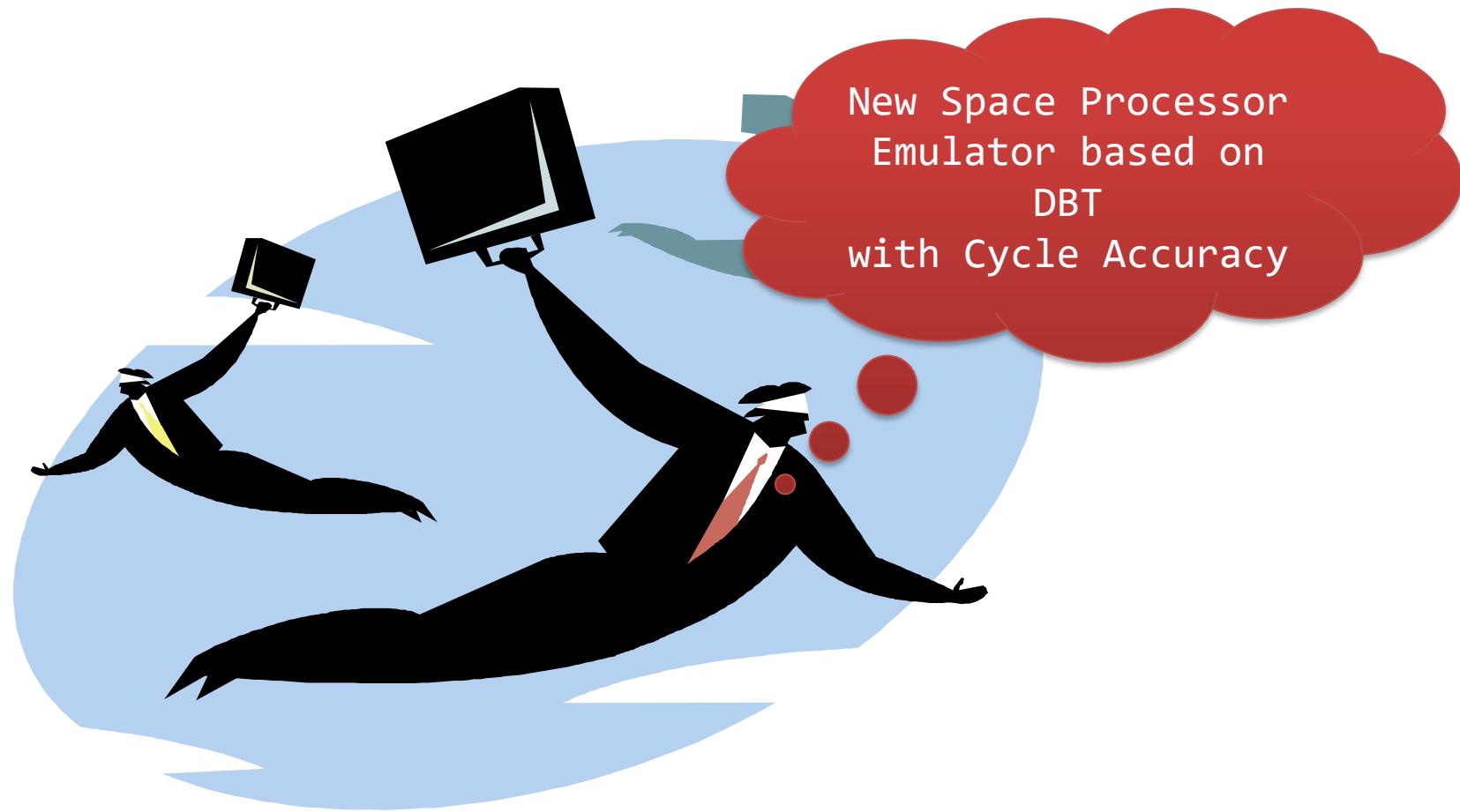
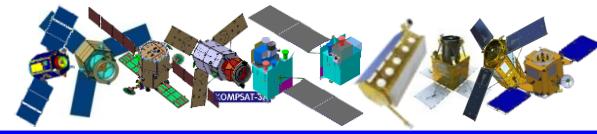


- QEMU laysim-erc32/leon3
 - Dynamic Binary Translation Emulator based on QEMU
 - Behavior modeling
 - *Not Cycle Accuracy! But high-performance*
- ➔ QEMU relies on host clock without instruction cycle counting
- ➔ Lead issues with guest clock and I/O timing
- ➔ It is not easy to run Flight Software which have time constrained attributions
- ➔ QEMU laysim-erc32/leon3 only run at Linux Machine

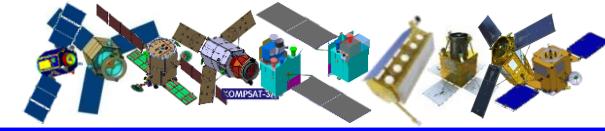


Final Solution

How to ?

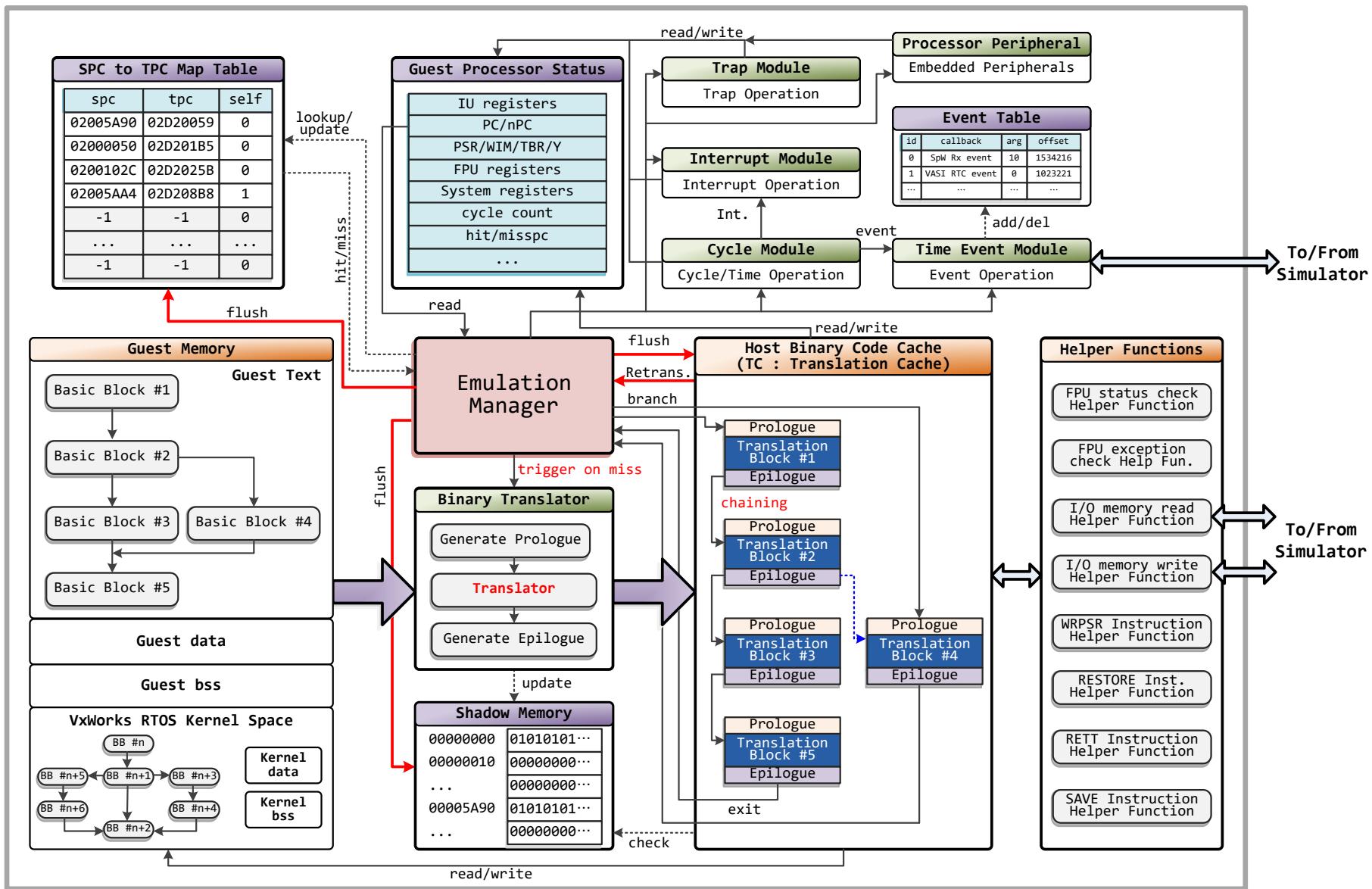
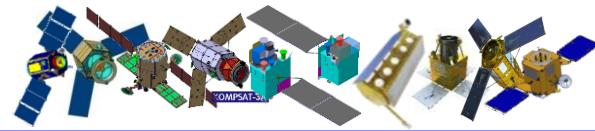


Final Solution

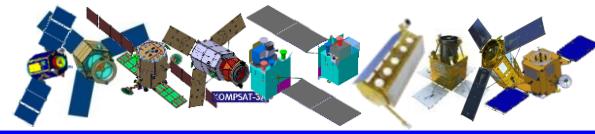


- Develop new Processor Emulator & Simulator from scratch
- Improve performance of Space Processor Emulator & Simulator
 - By Dynamic Binary Translation (DBT)
 - DBT engine generate from guest BB (ERC32) to host TB (Translation Block, x86) without IR (Intermediate Representation)
 - Minimize helper functions (C routine called from Translation Cache [TC])
 - Maximize chaining without self-chaining
 - By Multi-Thread : another avionics simulator run on another core
- Cycle Accurate
 - By cycle counting with static cycle-approximate model as chunk
 - By precise trap supporting
 - By fast interrupt handling
- Self-Modifying Code
 - Use **shadow memory** to resolve side effect of using *VirtualProtect()* in Windows

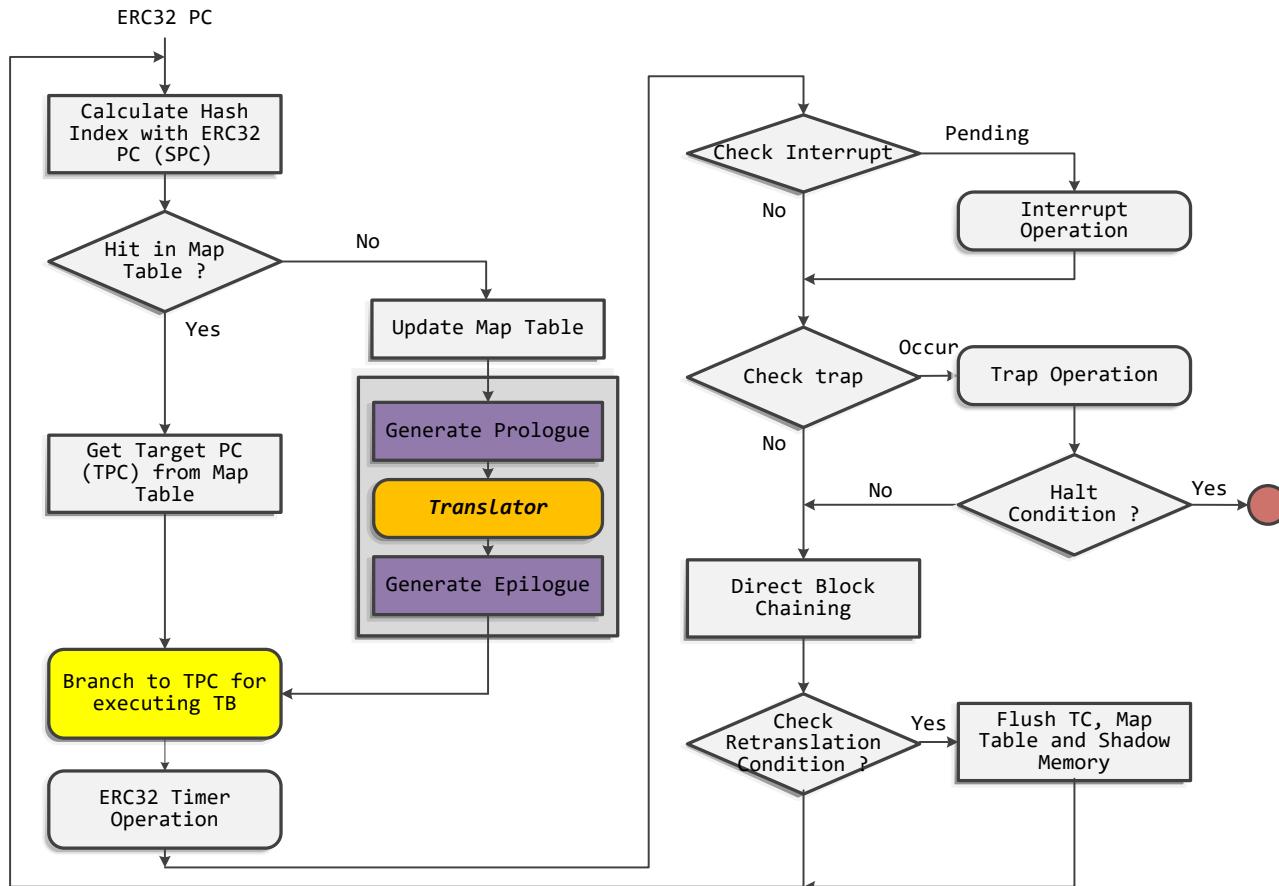
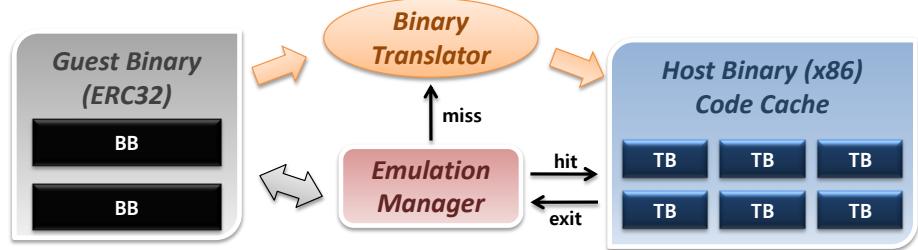
laysim-DBT for ERC32 [1/2]



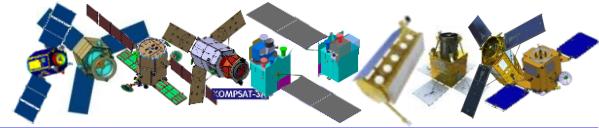
laysim-DBT for ERC32 [2/2]



- Emulation Manager
 - Do not use interpreter
 - Generate native host code directly using DBT

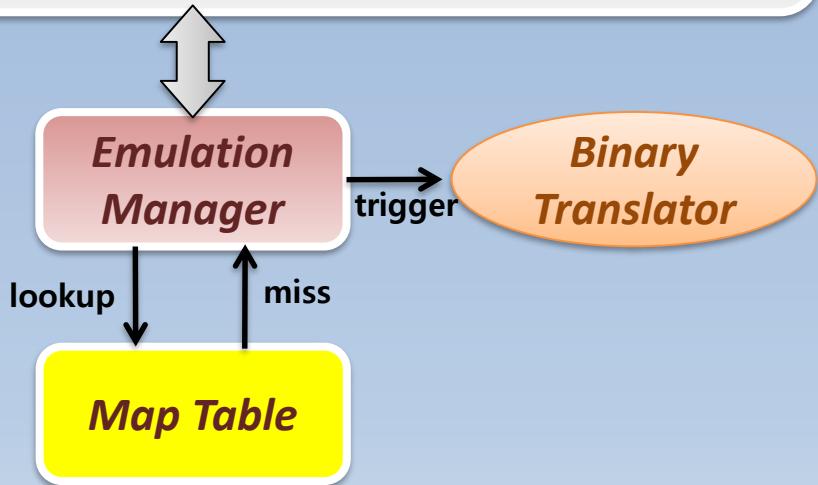


Translation Example [1/2]



Basic Block of ERC32

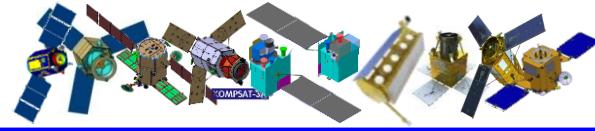
```
02000000    a0100000  clr    %10
02000004    29008004  sethi   %hi(0x2001000), %14
02000008    81c52000  jmp    %14 + 0x0
0200000c    01000000  nop
```



laysim-DBT for ERC32

```
ERC32 DBT Engine
#----- prologue -----
x86_emit_push_r32(edbp)
x86_emit_mov_r32_r32(esp, ebp)
x86_emit_mov_imm32_edx((guint32)(&ERC32))
#
#----- or -----
x86_emit_mov_m32_r32(edx, eax, (4 * op3_rs1))
x86_emit_mov_m32_r32(edx, ebx, (4 * op3_rs2))
x86_emit_or_r32_r32(ebx, eax)
x86_emit_mov_r32_m32(eax, edx, (4 * op3_rd))
#
#----- sethi -----
x86_emit_mov_imm32_eax((INST.inst_f2_sethi.imm22 << 10))
x86_emit_mov_r32_m32(eax, edx, (4 * INST.inst_f3_i0.rd))
#
#----- jmp1 -----
x86_emit_write_imm32_erc32_pc(pERC32->erc32.pc)
x86_emit_write_imm32_erc32_npc(pERC32->erc32.npc)
x86_emit_mov_m32_r32(edx, eax, (4 * op3_rs1))
x86_emit_mov_imm32_ebx(operand)
x86_emit_add_r32_r32(ebx, eax)
x86_emit_test_imm8_al(0x3)
x86_emit_jz_with_offset(22)
x86_emit_write_erc32_trap_with(ERC32_TRAP_07_MEM_NOT_ALIGN)
x86_emit_increas_icnt((inst_cnt+1))
x86_emit_return
x86_emit_mov_imm32_ebx(pERC32->erc32.pc)
x86_emit_mov_r32_m32(ebx, edx, (4 * op3_rd))
x86_emit_write_eax_erc32_npc
#
#----- epilogue -----
x86_emit_read_erc32_npc_eax
x86_emit_write_eax_erc32_pc
x86_emit_add_imm8_r32(4, eax)
x86_emit_write_eax_erc32_npc
x86_emit_increase_icnt(inst_cnt)
x86_emit_return
```

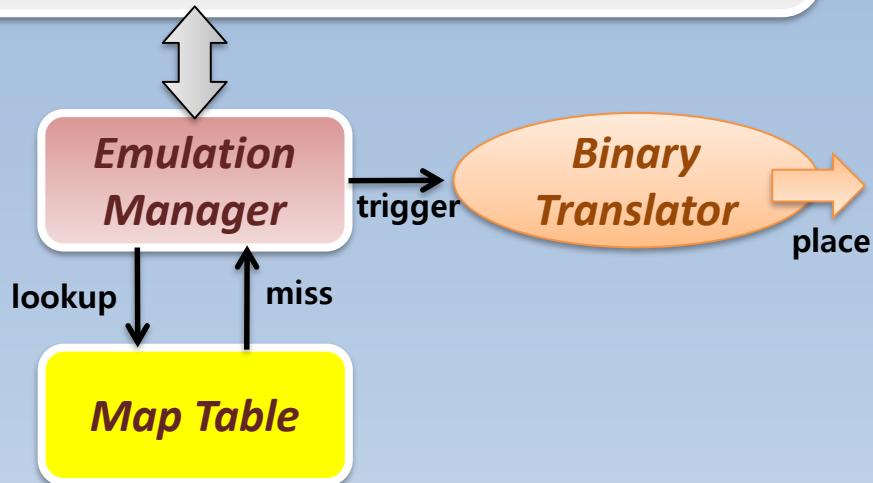
Translation Example [2/2]



Basic Block of ERC32

```

02000000    a0100000    clr    %10
02000004    29008004    sethi   %hi(0x2001000), %14
02000008    81c52000    jmp    %14 + 0x0
0200000c    01000000    nop
  
```



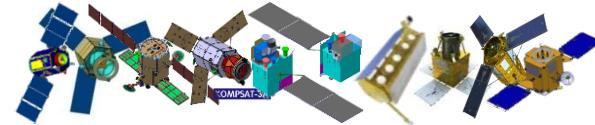
laysim-DBT for ERC32

x86 Translation Block in TC

```

#----- prologue -----
push %ebp
mov %esp, %ebp
mov $0x101b0e0,%edx ; ERC32 reg. address -> %edx
#----- or -----
mov 0x0(%edx), %eax ; ERC32.g0 -> %eax
mov 0x0(%edx), %ebx ; ERC32.g0 -> %ebx
or %ebx, %eax ; %ebx OR %eax -> %eax
mov %eax, 0x40(%edx) ; %eax -> ERC32.10
#----- sethi -----
mov $0x02001000, %eax ; 0x02001000 -> %eax
mov %eax, 0x50(%edx) ; %eax -> ERC32.14
#----- jmp1 -----
movl $0x2000008,0x280(%edx) ; Current PC -> ERC32.PC
movl $0x200000c,0x284(%edx) ; Current nPC -> ERC32.nPC
mov 0x50(%edx),%eax ; ERC32.14 -> %eax
mov $0x0,%ebx ; 0x00000000 -> %ebx
add %ebx,%eax ; %ebx ADD %eax -> %eax
test $0x3,%al ; Check memory aligned
je 0x191b553 ; Jump to JMPL operation
movl $0x7,0x3dc(%edx) ; 0x00000007 -> ERC32.trap
addl $0x3,0x3cc(%edx) ; Add 3 to ERC32.icnt
leave ; Return to Emulation Manager
ret
mov %eax,0x284(%edx) ; %eax -> ERC32.nPC
#----- epilogue -----
mov 0x284(%edx),%eax ; Read ERC32.nPC
mov %eax,0x280(%edx) ; ERC32.nPC -> ERC32.PC
add $0x4,%eax ; Add 0x4 to %eax
mov %eax,0x284(%edx) ; %eax -> ERC32.nPC
addl $0x5,0x3cc(%edx) ; Add 5 to ERC32.icnt
leave ; Return to Emulation Manager
ret
  
```

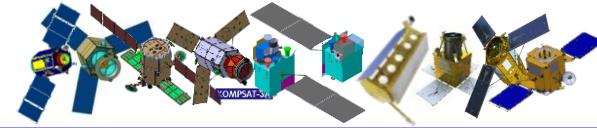
Emulator Performance Comparison



Benchmark	Intel Core i7-870 @ 2.93GHz (Windows 7 Enterprise 64bit)						Intel i7-6700K @ 4GHz laysim-erc32 DBT
	ERC32 CPU (20MHz)	LEON2-FT CPU (80MHz)	TSIM (Interpreter)	laysim-erc32 (Interpreter)	QEMU-laysim-erc32 (DBT)	laysim-erc32 DBT (DBT)	
Dhrystone (Iterations 800K)	11.67 MIPS	46.27 MIPS	23.99 MIPS	27.98 MIPS	126.79 MIPS	218.45 MIPS	494.97 MIPS
Whetstone (Single Precision)	10.18 MOPS	38.44 MOPS	20.71 MOPS	24.39 MOPS	110.82 MOPS	174.49 MOPS	286.02 MOPS
Whetstone (Double Precision)	9.57 MOPS	38.10 MOPS	20.67 MOPS	23.62 MOPS	113.53 MOPS	172.94 MOPS	271.67 MOPS
CoreMark (Iterations 60)	12.88 MIPS	52.75 MIPS	25.10 MIPS	29.79 MIPS	133.87 MIPS	328.76 MIPS	526.10 MIPS

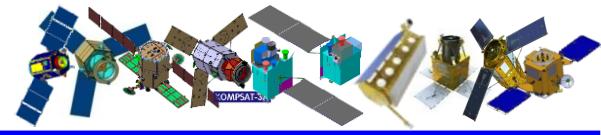
- All Benchmark program are compiled from LECC and BCC with -O0 optimization
- T-EMU 2.0 from TERMA shows 110MIPS on a 3.5GHz PC when running Dhrystone (from website)
- QERx from SCISYS shows 12.1x faster than TSIM (from paper)

Cycle Accuracy Evaluation



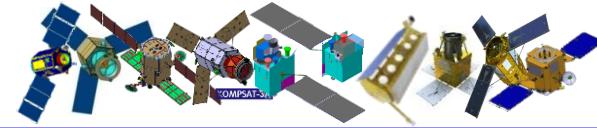
Benchmarks	Total Cycles and Cycle Accuracy				
	TSIM-ERC32	laysim-erc32		laysim-DBT	
Dhrystone	633,639,362	633,639,223	99.99%	604,838,274	95.45%
Whetstone SP	2,092,189,460	2,064,362,894	98.67%	2,261,798,805	108.11%
Whetstone DP	2,443,432,999	2,450,234,938	100.28%	2,626,116,296	107.48%
Stanford	37,118,286	36,898,680	99.41%	35,504,199	95.65%
CoreMark	286,125,763	286,125,297	99.99%	276,845,452	96.76%
MiBench.qsort	103,742,892	94,299,659	90.89%	104,063,427	100.31%
MiBench.dijkstra	399,444,134	399,442,953	99.99%	381,148,771	95.42%
MiBench.basicmath	292,667,095	320,161,268	109.39%	281,889,706	96.32%

- The cycle accuracy of laysim-DBT is **95.49% with ±8% error margin** against TSIM-ERC32 which is a cycle-true emulator of ERC32



Current laysim

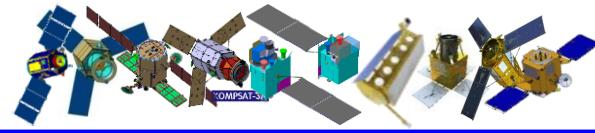
laysim Support Devices



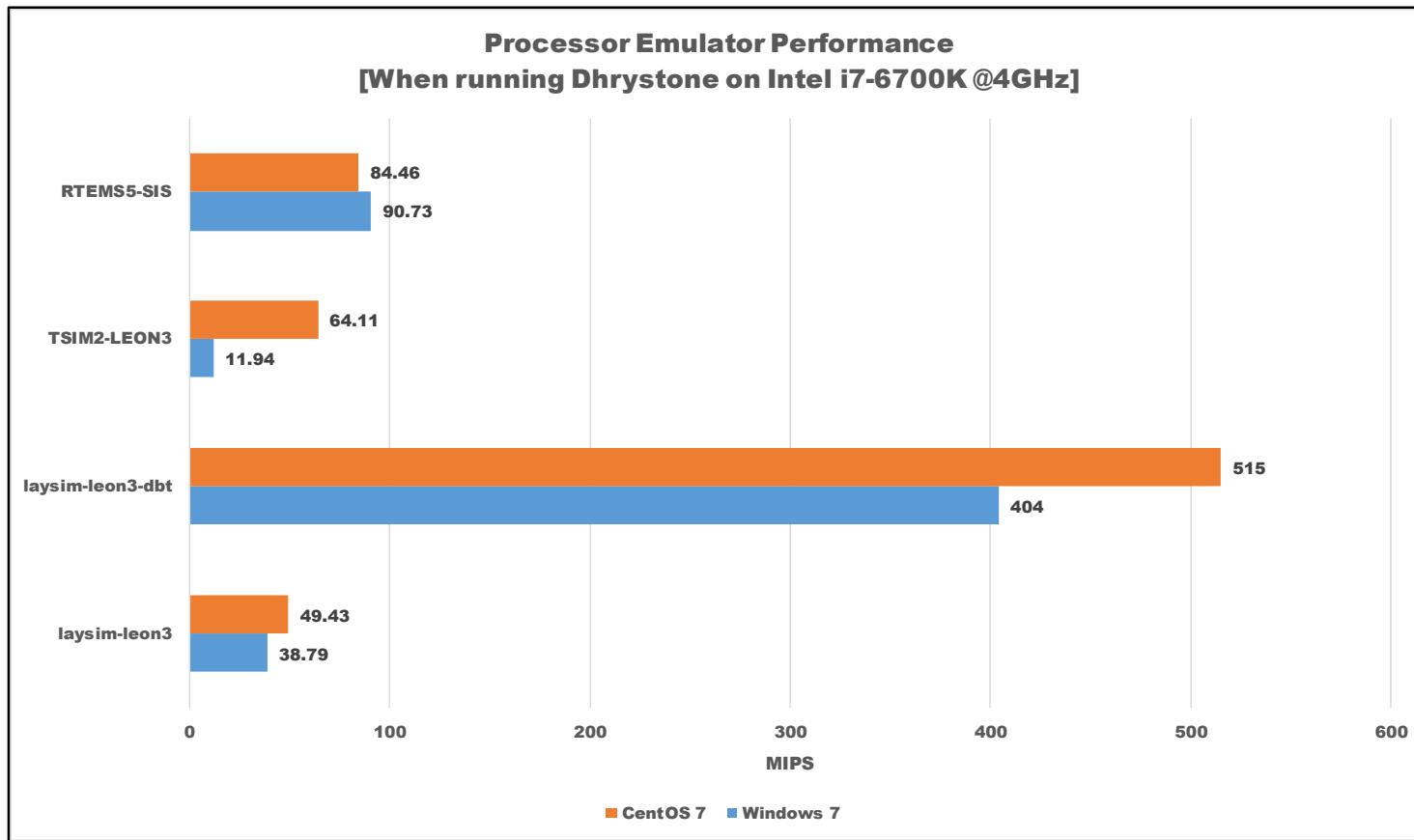
- Processors : ERC32, AT697F, LEON3, LEON4, UT700, GR712RC, GR740
- GRLIB : GPTIMER, IRQMP, FTMCTRL, AHBSTAT, and more
- KARILIB : CAN Master, DMAUART, TMU/TCU, and more

ERC32	AT697F	LEON3 & FT (0x01:0x003) (0x01:0x053)	LEON4-FT (0x01:0x048)	UT700	GR712RC	GR740
GPTIMER (0x01:0x011)	GPTIMER (0x01:0x011)	GRCLKGATE (0x01:0x02C)	GRTIMER (0x01:0x038)	IRQMP (0x01:0x00D)	IRQ(A)MP (0x01:0x00D)	MCTRL (0x04:0x00F)
FTMCTRL (0x01:0x054)	AHBSTAT (0x01:0x052)	AHB2AHB (0x01:0x020)	AHBCTRL	APBCTRL (0x01:0x006)	GRIOMMU (0x01:0x04F)	GRPCI2 (0x01:0x07C)
FTAHBRAM (0x01:0x050)	L2CACHE (0x01:0x04B)	APBUART (0x01:0x00C)	CAN_OC (0x01:0x019)	GRSPW2 (0x01:0x029)	GRSPWROUTER (0x01:0x03E)	GRETH (0x01:0x01D)
GR1553B (0x01:0x04D)	GRGPIO (0x01:0x01A)	GRGPREG (0x01:0x087)	B1553BRM (0x01:0x072)	Actel 1553BRM	LAN91C	KARI RTC
KARI CAN Master	KARI CAN Slave	KARI DMAUART	KARI MDMAUART	KARI TMU	KARI TCU	KARI Mass Memory
KARI BIOM	KARI PIOM	KARI CIOM	KARI PDM	KARI RU	KARI RU	KARI PCDU

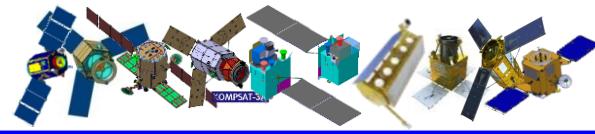
laysim Performance



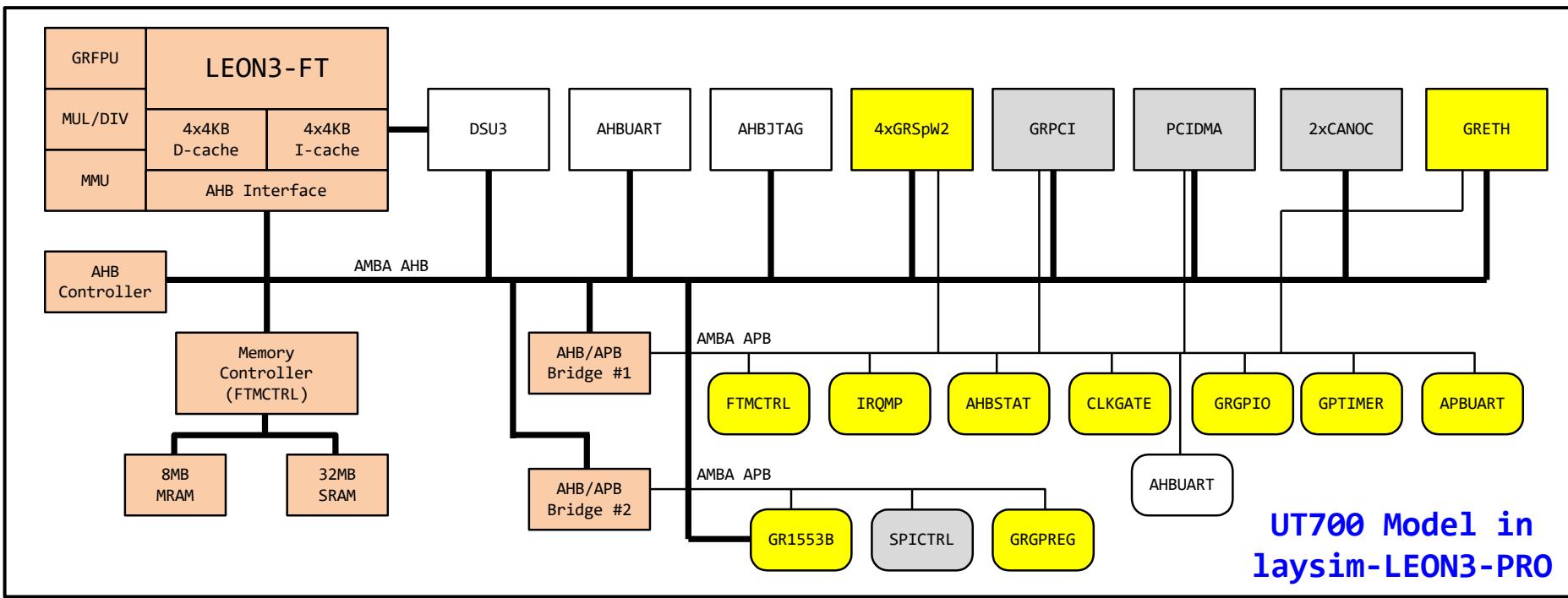
- Run Dhrystone for LEON3 (Single Core) at 100MHz
 - Emulator Performance with laysim-leon3, laysim-leon3-dbt, TSIM2-LEON3 and RTEMS5-SIS
 - RTEMS5-SIS does not emulate Cache System of LEON3
 - laysim-leon3-dbt shows outstanding performance (over 515MIPS on CentOS 7)



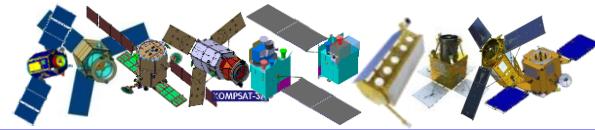
laysim-UT700 [1/2]



- UT700 processor with 100MHz clock
 - LEON3-FT and GRFPU
 - 16KB 4-way I-CACHE and 16KB 4-way D-CACHE
 - Many UT700 GRLIB IP cores are emulated
 - No GRPCI, PCIDMA and SPICTRL emulation
 - 8MB MRAM, 16MB SDRAM and test I/O
 - Also provide loadable modules for laysim-LEON3-PRO



laysim-UT700 [2/2]



laysim for UT700 processor (interpreter) v0.99 for API by layright

Control UT700 Source About

UT700 REGS FPU REGS APB REGS cache UART-A I/O

Failed to initialize grPCI driver -2

--- BUS TOPOLOGY ---

```
|--> DEV 0x5f8de570 GAISLER_LEON3FT
|--> DEV 0x5f8de5d8 GAISLER_AHBUART
|--> DEV 0x5f8de640 GAISLER_AHBJTAG
|--> DEV 0x5f8de6a8 GAISLER_PCIFBRG
|--> DEV 0x5f8de710 GAISLER_DMACTRL
|--> DEV 0x5f8de778 GAISLER_ETHMAC
|--> DEV 0x5f8de7e0 GAISLER_SFW2
|--> DEV 0x5f8de848 GAISLER_SFW2
|--> DEV 0x5f8de8b0 GAISLER_SFW2
|--> DEV 0x5f8de918 GAISLER_SFW2
|--> DEV 0x5f8de980 GAISLER_GR1553B
|--> DEV 0x5f8de9e8 GAISLER_FTMCTRL
|--> DEV 0x5f8dea50 GAISLER_APBMST
|--> DEV 0x5f8deab8 GAISLER_LEON3DSU
|--> DEV 0x5f8deb20 GAISLER_APBMST
|--> DEV 0x5f8deb88 GAISLER_CANAHB
|--> DEV 0x5f8deb00 GAISLER_CANAHB
|--> DEV 0x5f8dec48 GAISLER_APBUART
|--> DEV 0x5f8decb0 GAISLER_IRQMP
|--> DEV 0x5f8ded18 GAISLER_GPTIMER
|--> DEV 0x5f8ded80 GAISLER_CLKGATE
|--> DEV 0x5f8dede8 ESA_PCIARB
|--> DEV 0x5f8dee50 GAISLER_GPIO
|--> DEV 0x5f8deeb8 GAISLER_AHBJSTAT
|--> DEV 0x5f8def20 GAISLER_SPICTRL
|--> DEV 0x5f8def88 GAISLER_GPREG
```

You can use the shell commands drvmgr and more about the system

Creating /etc/passwd and group with three root/pwd , test/pwd, rtems/NO PASSWORD

RTEMS SHELL (Ver.1.0-FRC):dev/console. Dec [/] #

laysim Control Console

```
laysim for UT700 processor (interpreter) v0.99 for API by layright
laysim>
No MRAM file
SRAM file is loaded - C:\MinGW\msys\1.0\home\jwchoi\[400] New LEON3 Emulator\UT700\UT700-Interpreter\v1.0-Formal\RCC-test\rtems-shell.elf
- .text,    addr: 0x5F800000, size 546768 bytes
- .data,    addr: 0x5F8857D0, size 11328 bytes
- .jcr,     addr: 0x5F888410, size 4 bytes
- .bss,     addr: 0x5F888420, size 16608 bytes (not loaded, just info.)
- found DWARF debug information
```

UT700 RAM file is loaded, starts from RAM

```
laysim> run
laysim> stop
laysim-ut700 emulator is stopped by user
```

laysim> reg

	INS	LOCALS	OUTS	Globals
0:	00000000	5F88ED28	00000000	00000000
1:	00000000	5F88B400	5F88B400	5F80A3DC
2:	00000000	01000000	5F880100E	00000000

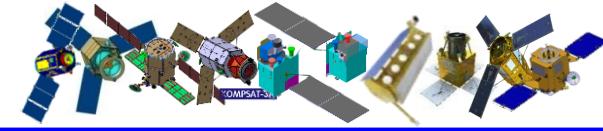
laysim Source Code Window

File Name : /opt/rtems-4.10-mingw/src/rtems-4.10/c/src/lib/libbsp/sparc/leon3/startup/bspidle.S

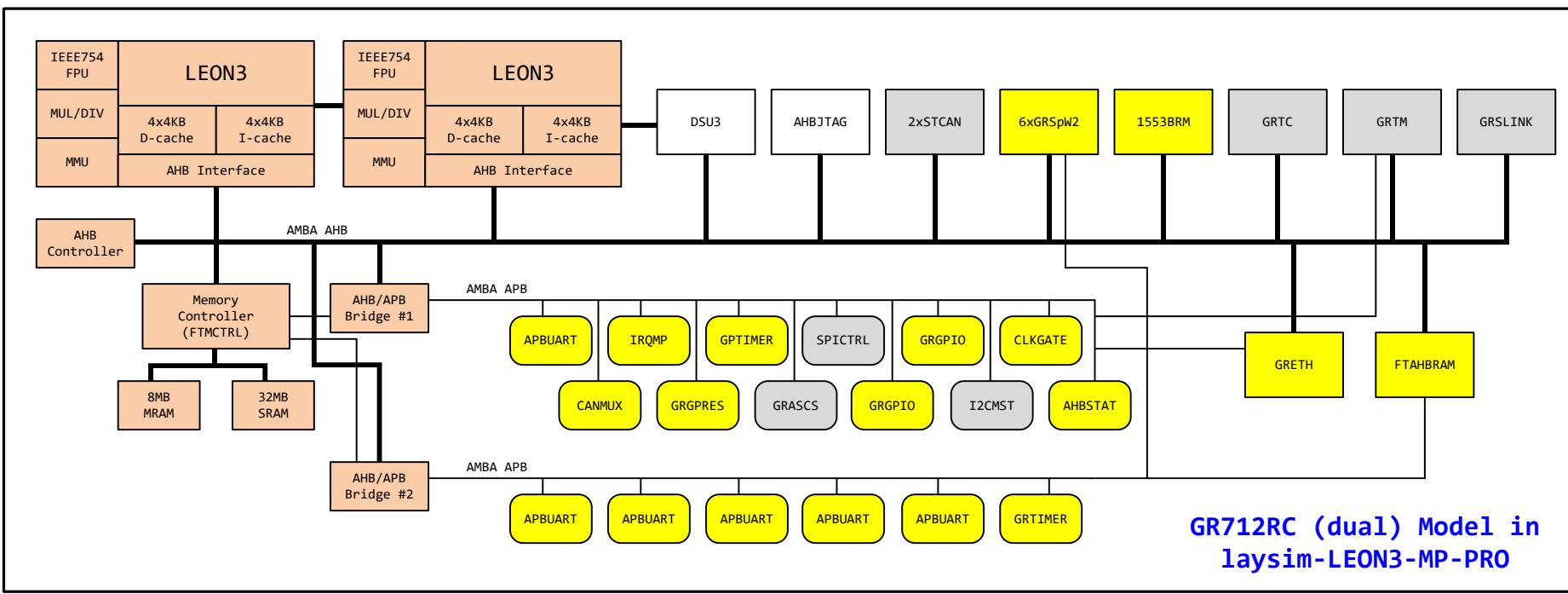
```
5F80A3DC A7800000 wr %g0, %g0, %asrl9
> 5F80A3E0 C0838020 lda [%sp] 0x1, %g0
5F80A3E4 10BFFF FE ba 0x5f80a3dc
5F80A3E8 01000000 nop
5F80A3EC 81C3E008 jmpl %o7 + 0x8, %g0
5F80A3F0 01000000 nop
```

```
1  /*
2   * Idle Thread Body
3   *
4   * This routine puts LEON3 in power-down mode
5   *
6   * COPYRIGHT (c) 2004.
7   * Gaisler Research.
8   *
9   * The license and distribution terms for this
10  * found in the file LICENSE in this distribution.
11  * http://www.rtems.com/license/LICENSE.
12  *
13  *
14  * $Id$
15  */
16
17
18
19  #include <rtems/asm.h>
20
21  /* LEON specific power-down function */
22
23      .align 4
24  PUBLIC(bsp_idle_thread):
25  SYM(bsp_idle_thread):
26  pwdloop:    mov    %g0, %asrl9
27          lda    [%sp] 1, %g0
28          ba    pwdloop
! Need
```

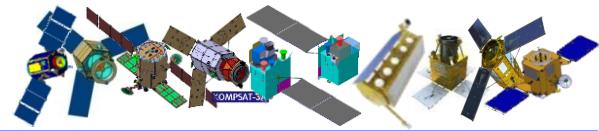
laysim-GR712RC [1/2]



- GR712RC processor with 100MHz clock
 - Dual LEON3-FT and GRFPU
 - 16KB 4-way I-CACHE and 16KB 4-way D-CACHE
 - Many GR712RC GRLIB IP cores are emulated
 - No STCAN, GRTC, GRTM, GRSLINK, GRASCS, I2CMST and SPICTRL emulation
 - 8MB MRAM, 8MB SRAM and test I/O
 - Also provide loadable modules for laysim-LEON3-PRO & laysim-LEON3-MP-PRO



laysim-GR712RC [2/2]



laysim for GR712RC Dual Core Processor (CACHE) v1.2 by layright

Control CORE#0 CORE#1 Traffic About

GR712RC UART0

```

Start RESULT list processing.
Waiting until list processed
List processed.

Response to message 1: (len: 8, tswl: 800, tsw2: 0)
  0x20 .. 0x21 .. 0x22 .. 0x23 .. 0x24 .. 0x25 .. 0x26 ..
Response to message 2: (len: 8, tswl: 800, tsw2: 0)
  0x40 .. 0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)
Response to message 3: (len: 8, tswl: 800, tsw2: 0)
  0x60 .. 0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)
Response to message 4: (len: 9, tswl: 800, tsw2: 0)
  0x80 .. 0x81 .. 0x82 .. 0x83 .. 0x84 .. 0x85 .. 0x86 ..

Sleeping 15s
----- BC: START LIST EXECUTION -----
Start CMD list processing.
Waiting until list processed
List processed.
Sleeping 20s
----- BC: START LIST EXECUTION -----
Start RESULT list processing.
Waiting until list processed
List processed.

Response to message 1: (len: 8, tswl: 800, tsw2: 0)
  0x20 .. 0x21 .. 0x22 .. 0x23 .. 0x24 .. 0x25 .. 0x26 ..
Response to message 2: (len: 8, tswl: 800, tsw2: 0)
  0x40 .. 0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)
Response to message 3: (len: 8, tswl: 800, tsw2: 0)
  0x60 .. 0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)
Response to message 4: (len: 9, tswl: 800, tsw2: 0)
  0x80 .. 0x81 .. 0x82 .. 0x83 .. 0x84 .. 0x85 .. 0x86 ..

Sleeping 15s

```

laysim Control Console

```

laysim>
No MRAM file
CORE#0 SRAM file is loaded - C:\MinGW\msys\1.0\home\jwchoi\[270] laysim-GR712RC\GR712RC-Interpreter\v1.2-CASA-FIX\Examples\RCC-1.2.23\rtems_brm_bc.elf
- .text, addr: 0x40000000, size 203888 bytes
- .data, addr: 0x40031C70, size 6416 bytes
- .jcr, addr: 0x40033580, size 4 bytes
- .bss, addr: 0x40033590, size 5248 bytes (not loaded, just info.)
- found DWARF debug information

```

GR712RC 1553B Monitor

(BUS A) : BC(GR712RC) -> RT		Msg No : 18 [7017351744]
Cmnd:	0868 (01-R-03-08)	RT1-SA03
Data:	0000 0047 0052 0009 0000 0000 0000 0000	
Status:	0800	
(BUS A) : BC(GR712RC) -> RT		Msg No : 19 [7017372495]
Cmnd:	0889 (01-R-04-09)	RT1-SA04
Data:	0000 0047 0052 000A 0000 0000 0000 0000	
Status:	0800	
(BUS A) : RT -> BC(GR712RC)		Msg No : 20 [9018335238]
Cmnd:	0C28 (01-T-01-08)	RT1-SA01
Status:	0800	
Data:	0020 0021 0022 0023 0024 0025 0026 0027	
(BUS A) : RT -> BC(GR712RC)		Msg No : 21 [9018355988]
Cmnd:	0C48 (01-T-02-08)	RT1-SA02
Status:	0800	
Data:	0040 0041 0042 0043 0044 0045 0046 0047	
(BUS A) : RT -> BC(GR712RC)		Msg No : 22 [9018376738]
Cmnd:	0C68 (01-T-03-08)	RT1-SA03

GR712RC 1553B Overall Traffic

RT#1 :	A	B	E
RT#2 :	A	B	E
RT#3 :	A	B	E
RT#4 :	A	B	E
BRD (RT31) :	A	B	<input type="button" value="Clear"/>

1553B Receive Editor

Remote Terminal : RT#1

Sub-Address : 01 Apply

Status Word : 0800 No Response

1553B Transmit Editor

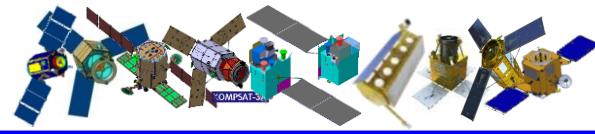
Remote Terminal : RT#1

Sub-Address : 01 Apply

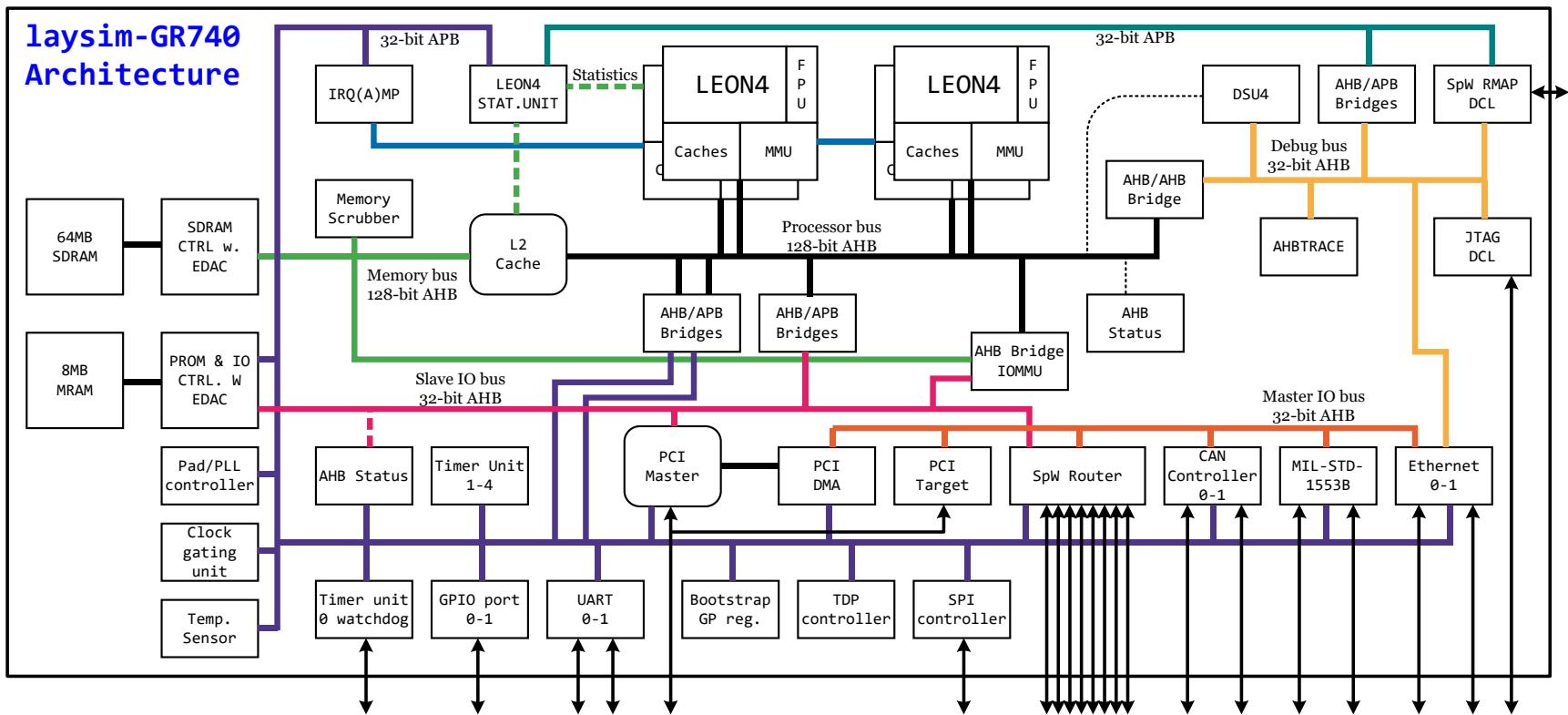
Status Word : 0800 No Response

00 - 03	0020	0021	0022	0023
04 - 07	0024	0025	0026	0027
08 - 11	0028	0029	002A	002B

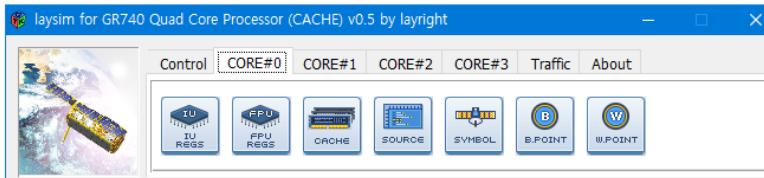
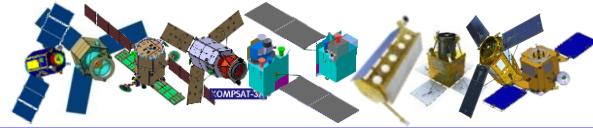
laysim-GR740 [1/3]



- GR740 processor with 250MHz clock
 - Quad LEON4-FT and GRGPU
 - 16KB 4-way I-CACHE, 16KB 4-way D-CACHE and 2MB Level-2 Cache
 - Many GR740 GRLIB IP cores are emulated
 - Under development for GRCAN, GRIOMMU and TDP emulation
 - 8MB MRAM and 64MB SDRAM



laysim-GR740 [2/3]



```

laysim for GR740 Quad Core Processor (CACHE) v0.5 by layright
Control CORE#0 CORE#1 CORE#2 CORE#3 Traffic About

GR740 UART0

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.lfea3c7a66f062c6e4cledc4c75dcc3728602e9f9-modified
*** TEST STATE: EXPECTED-PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.2.0

CPU 3 running task Init - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA01 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA05 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA06 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA07 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA08 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA09 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA10 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA11 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 2 running task TA12 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 3 running task TA04 - rtems_clock_get_tod 09:00:00 12/31/1988
CPU 3 running task TA01 - rtems_clock_get_tod 09:00:05 12/31/1988
CPU 3 running task TA01 - rtems_clock_get_tod 09:00:10 12/31/1988
CPU 3 running task TA02 - rtems_clock_get_tod 09:00:10 12/31/1988
CPU 3 running task TA01 - rtems_clock_get_tod 09:00:15 12/31/1988
CPU 3 running task TA03 - rtems_clock_get_tod 09:00:15 12/31/1988
CPU 3 running task TA01 - rtems_clock_get_tod 09:00:20 12/31/1988
CPU 3 running task TA02 - rtems_clock_get_tod 09:00:20 12/31/1988
CPU 3 running task TA04 - rtems_clock_get_tod 09:00:20 12/31/1988
CPU 3 running task TA05 - rtems_clock_get_tod 09:00:25 12/31/1988
CPU 3 running task TA01 - rtems_clock_get_tod 09:00:25 12/31/1988
CPU 3 running task TA06 - rtems_clock_get_tod 09:00:30 12/31/1988
CPU 0 running task TA01 - rtems_clock_get_tod 09:00:30 12/31/1988
CPU 0 running task TA02 - rtems_clock_get_tod 09:00:30 12/31/1988
CPU 0 running task TA03 - rtems_clock_get_tod 09:00:30 12/31/1988

*** END OF TEST SMP 8 ***

```

laysim Control Console

```

CORE#0 SDRAM file is loaded - C:\MinGW\msys\1.0\home\jwchoi\[290] laysim-gr740\GR740-Interpreter\v0.5-Major\Exam-
uites\smptests\smp08\smp08.elf
- .text, addr: 0x00000000, size 212320 bytes
- .rtemsroset, addr: 0x00033D60, size 112 bytes
- .rtemsstack, addr: 0x00033E00, size 16384 bytes (not loaded, just info.)
- .data, addr: 0x00037E00, size 4944 bytes
- .bss, addr: 0x00039180, size 29536 bytes (not loaded, just info.)
- found DWARF debug information
No CORE#1 SDRAM file
Only GR740 CORE#0 SDRAM file is loaded, starts from SDRAM
Set CORE#0 %sp/%fp : 0x03FFFFFF/0x04000000

laysim> run
laysim>
CORE#1 is started by MSR
CORE#2 is started by MSR
CORE#3 is started by MSR
CORE#0 halts because of Error Mode [8808995543cycles/25.4291sec]

laysim> stop
laysim-gr740 emulator is stopped by user

laysim> core 4
GR740 all COREs are selected for debugging

laysim> reg

----- CORE#0 -----
INS      LOCALS      OUTS      GLOBALS
0: 00000005  F3000FC1  00000005  00000000
1: 00000000  0001C670  00000004  00000001
2: 00038AC8  0001C674  00000000  00000005
3: 00000004  00000080  FF904000  00000000
4: 0003ED10  0001C400  0000000E  00000000
5: 80000000  00000010  00000004  00000000
6: 0004C238  0003D738  0004C1C0  0003F3C0
7: 0002C038  0003BBC8  00011A04  00000000

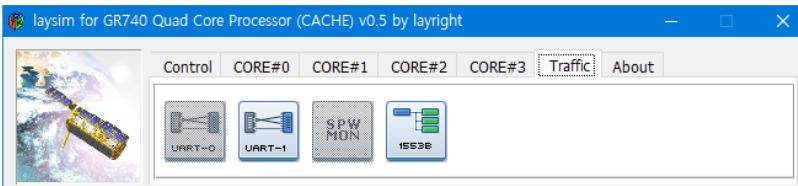
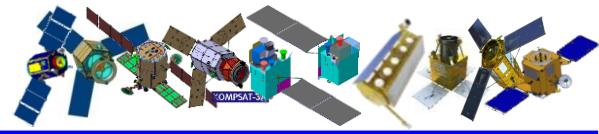
----- CORE#1 -----
INS      LOCALS      OUTS      GLOBALS
0: 0000000A  F3400FC0  0003F6A4  00000000
1: 00000000  000116F4  00035C64  00001A30
2: 00000005  000116F8  00007406  00000003
3: 00000DC3  F34000C0  0003A128  0A010001
4: 0003F6A0  00000100  0003A12C  000025BC
5: 00035C64  0003A0E4  00000001  00000078
6: 00035C00  0003D738  00035BA0  0003F5C0
7: 00011A04  0003A318  00001B24  00000000

psr:F3000FC1 wim:00000010 tbr:00000800 y:00000012
[n:0 z:0 v:0 c:0 ef:0 pil:15 s:1 ps:1 et:0 cwp:1]
psr:F3400EE0 wim:00000080 tbr:000001E0 y:00000000
[n:0 z:1 v:0 c:0 ef:0 pil:14 s:1 ps:1 et:1 cwp:0]

pc: 0001C660 : 91D02000  ta      0x0
npc: 0001C664 : 82102001  or      %g0, 0x1, %gl
pc: 00001DCC : C0838020  lda     [%sp] 0x1, %g0
npc: 00001DD0 : 30BFFFFE  ba,a   0x00001dc8

```

laysim-GR740 [3/3]



GR740 UART

Note that the spw-links will not be used.
There are 4 GRSPW cores in the system.
System clock: 10000 us / tick

Starting GRSPW0: DMA Started Successfully
NOTE: running on SPW-ROUTER DMA SpaceWire link (no link-state available)
RMAP connection is working
Initializing data array

Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000455 seconds
RMAP read cycle completed in 0.000459 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000455 seconds. WRITE RATE: 8790.498608 KiB/s
Read 4096 bytes from target in 0.000459 seconds. READ RATE: 8715.012451 KiB/s
Wrote 5 packets to target in 0.000455 seconds. WRITE RATE: 10988.123260 pkt/s
Read 5 packets from target in 0.000459 seconds. READ RATE: 10893.765563 pkt/s

EXAMPLE 1 SUCCESSFULLY COMPLETED.

Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000391 seconds
RMAP read cycle completed in 0.000150 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000391 seconds. WRITE RATE: 10231.569447 KiB/s
Read 4096 bytes from target in 0.000150 seconds. READ RATE: 26670.189369 KiB/s
Wrote 5 packets to target in 0.000391 seconds. WRITE RATE: 12789.461808 pkt/s
Read 5 packets from target in 0.000150 seconds. READ RATE: 33337.736711 pkt/s

EXAMPLE 2 SUCCESSFULLY COMPLETED.

laysim Control Console

```
laysim for GR740 Quad Core Processor (CACHE) v0.5 by layright
laysim>
No MRAM file
CORE#0 SDRAM file is loaded - C:\MinGW\msys\1.0\home\jwchoi\[290] laysim-gr740\GR740-Interpreter\v0.5-
rmap-logical-addressing.elf
- .text, addr: 0x00000000, size 206784 bytes
- .data, addr: 0x000327C0, size 7248 bytes
- .jcr, addr: 0x00034410, size 4 bytes
- .bss, addr: 0x00034420, size 156472 bytes (not loaded, just info.)
- found DWARF debug information
No CORE#1 SDRAM file
Only GR740 CORE#0 SDRAM file is loaded, starts from SDRAM
Set CORE#0 %sp/%fp : 0x03FFFFFF/0x04000000

laysim> run
laysim>
CORE#0 halts because of Error Mode [343406923cycles/15.1485sec]

GR740 all cores halt because of Error Mode
```

SpaceWire Traffic Monitor

```
(GR740 SpW#1 --> SpW Port 3 --> Node#3) [256598475]
42 01 4C 00 20 00 00 00 40 00 00 00 00 00 04 3D

(Node#3 --> AMBA Port 1 --> GR740 SpW#1) [256598864]
20 01 0C 00 42 00 00 00 00 04 26 00 00 00 00 00 00

(GR740 SpW#1 --> SpW Port 3 --> Node#3) [256604877]
42 01 6C 00 20 00 01 00 40 00 00 00 00 00 04 6B FF FF FF FF 7B

(Node#3 --> AMBA Port 1 --> GR740 SpW#1) [256605367]
20 01 2C 00 42 00 01 99

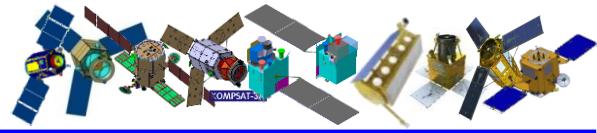
(GR740 SpW#1 --> SpW Port 3 --> Node#3) [256610022]
42 01 4C 00 20 00 02 00 40 00 00 00 00 00 04 65

(Node#3 --> AMBA Port 1 --> GR740 SpW#1) [256610412]
20 01 0C 00 42 00 02 00 00 00 04 35 FF FF FF FF 7B

(GR740 SpW#1 --> SpW Port 3 --> Node#3) [263816676]
42 01 6C 00 20 00 03 00 40 00 00 00 03 E0 2B 00 00 00 00 00
00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06
00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00
00 0C 00 00 00 0D 00 00 00 0E 00 00 00 0F 00 00 00 10 00 00 00 11
00 00 00 12 00 00 00 13 00 00 00 14 00 00 00 15 00 00 00 16 00 00
00 17 00 00 00 18 00 00 00 19 00 00 00 1A 00 00 00 1B 00 00 00 1C
```

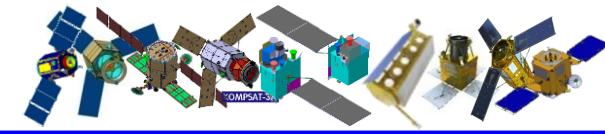
GRSpW2 Test

Node #1 No Response :	<input type="checkbox"/>	No Response
Node #1 Write Status :	<input type="checkbox"/>	Error Issue
Node #1 Read Status :	<input type="checkbox"/>	Error Issue
Node #1 Disconnect :	<input checked="" type="checkbox"/>	Error Issue
Node #2 No Response :	<input type="checkbox"/>	No Response
Node #2 Write Status :	<input type="checkbox"/>	Error Issue
Node #2 Read Status :	<input type="checkbox"/>	Error Issue
Node #2 Disconnect :	<input checked="" type="checkbox"/>	Error Issue
Node #3 No Response :	<input type="checkbox"/>	No Response
Node #3 Write Status :	<input type="checkbox"/>	Error Issue
Node #3 Read Status :	<input type="checkbox"/>	Error Issue
Node #3 Disconnect :	<input checked="" type="checkbox"/>	Error Issue
Node #4 No Response :	<input type="checkbox"/>	No Response
Node #4 Write Status :	<input type="checkbox"/>	Error Issue
Node #4 Read Status :	<input type="checkbox"/>	Error Issue
Node #4 Disconnect :	<input checked="" type="checkbox"/>	Error Issue
Traffic Clear :	<input type="button" value="Clear"/>	



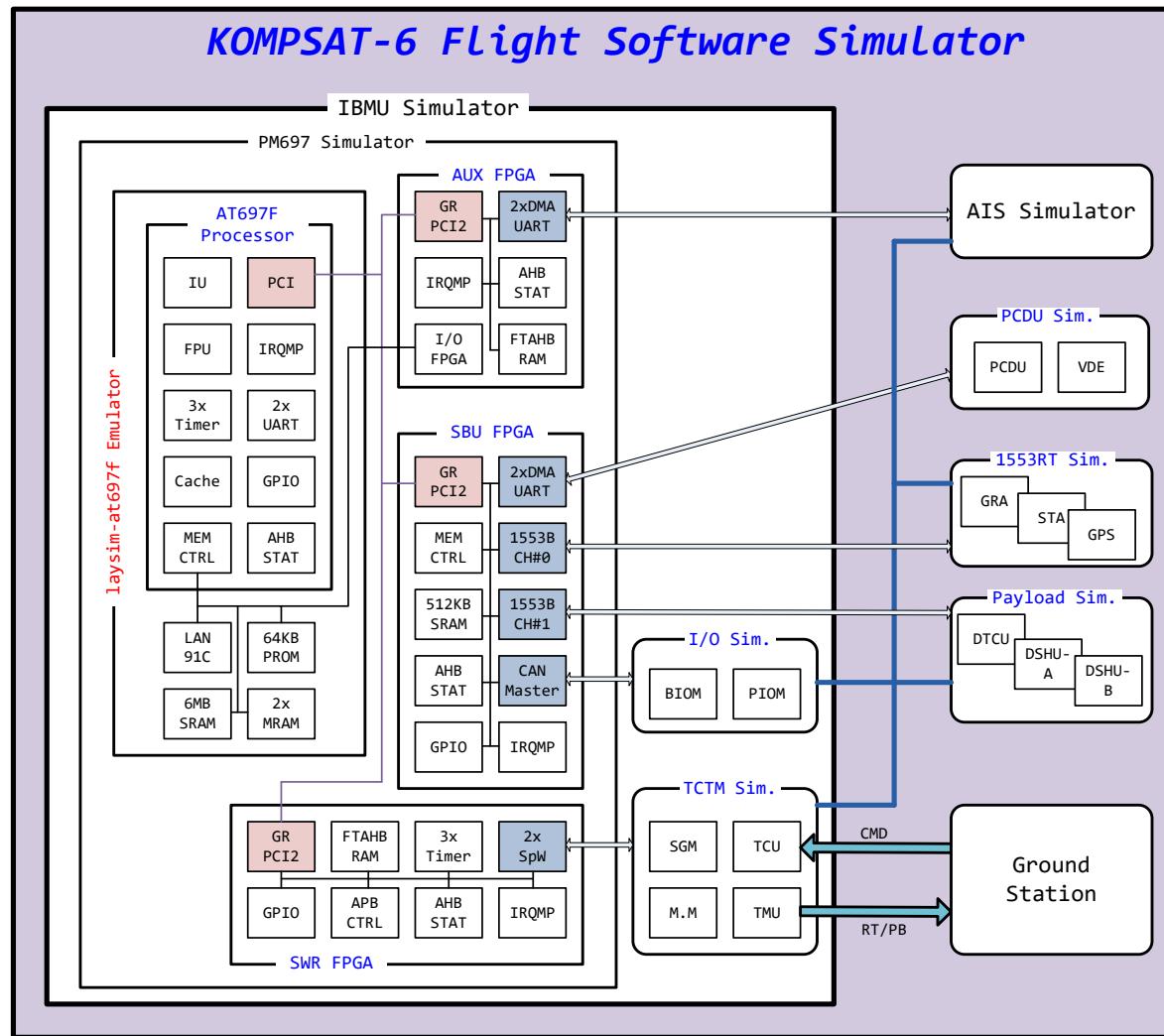
Integration laysim to KOMPSAT-FSS

KOMPSAT-6 FSS [1/2]

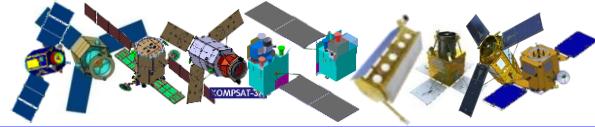


- K6-FSS Architecture

- laysim-AT697F → PM697 Simulator → IBMU Simulator → K6-FSS



KOMPSAT-6 FSS [2/2]



Command & Telemetry Interface

The Command & Telemetry Interface section contains several windows:

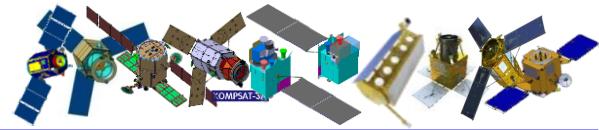
- SBU DMAUART #0 Monitor**: Shows RX and TX data frames.
- SBU DMAUART #0 Select**: Configuration for SBU DMAUART #0.
- SBU CAN Monitor**: Shows CAN bus activity.
- PM697 Control Console**: Displays log messages from the PM697 control console.
- AT697 Cache Window**: Shows memory dump of the AT697 cache.
- AT697 UART-A**: Shows the AT697 UART-A terminal window.
- Access Read/Write**: A stack of windows for reading and writing to memory locations.
- Virtual Network**: A stack of windows for managing network interfaces.

mini-IVF (Integrated Verification Facility) for K6 v0.9 by layright

The mini-IVF interface includes:

- A toolbar with icons for Teleport, RTTM, Commander, TLM Viewer, AIS Viewer, Event Viewer, Post Processing, ISL Manager, and Preferences.
- Teleport**: A window for managing teleports.
- Manual Command Manager**: A window for managing manual commands.
- Command Information**: A window showing command details.
- Memory**: A window showing memory dump.
- Tornado - Shell 127.0.0.1@LAYRIGHT-PC**: A terminal shell window.
- WindView**: A window for viewing memory and registers.
- Primary Mass Memory**: A window for managing primary mass memory.
- Wind River Systems**: A waveform viewer showing signal traces.

Full Dynamic KPLO Simulator



- KPLO-FSS + LUNASIM (Dynamic AOCS Simulator) + STK (Visualization) + ITOS (Ground Segment)

LUNASIM

```
cat_atm.08 guides_Teff.cat
Init Mode: Enter
Initialize models . . . OK
About to initialize visualization
Init UDP SOCK for FSS . . . DAS IP address set to default: 127.0.0.1
DAS port number to default: 28980
Init DAS UDP SOCK for RHEC . . . OK
Init DAS UDP SOCK for TAP . . . OK
Waiting Time Sync from FSS . . . Current time is = 0.00 sec
Nadir Command <Current_time = 1.00000>
Current time is = 1.00000
L_inf Norm Torque Method Enabled !!
Current time is = 1.00000
L_inf Norm Torque Method Enabled !!
Current time is = 1.00000
Disable Wheel Speed Limitation Method !!
ACS New Submode Set to 1. <Current_time = 1.00000>
Enable all Wheel Loops !!
ACS New Submode Set to 2 (TP), <Current_time = 9.00000>
Current time is = 1000.00 sec
Back to Nadir <Current_time = 1000.00000>
```

KPLO Flight Software Simulator v1.0 with LUNASIM (DBT Version) by layright

ITOS - Real Time Operation Manager

ITOS

kplo_TP1 - STK 9 - [3D Graphics 1 - Moon]

The image displays a complex software interface for a space mission. It includes multiple windows:

- LUNASIM**: Shows log output for initializing models, setting up UDP and DAS ports, and enabling various torque methods.
- KPLO Flight Software Simulator**: A control panel with buttons for PWR, OVERALL, and CONFIG, and tabs for Control, AT697F, About, and Telemetry.
- ITOS - Real Time Operation Manager**: Shows account information (Mission: KPLO, User: super), current time (KST 2020/05/14 08:59:52), and a control panel with buttons for RELEASE and CONFIG.
- ITOS**: A detailed real-time operation manager showing telemetry data for Rate (TAROLRATE, TAYAWRATE, TAPI TRATE) over time (13:46:59.000 to 13:51:59.000). The data shows significant spikes in the TAROLRATE and TAYAWRATE channels.
- kplo_TP1 - STK 9 - [3D Graphics 1 - Moon]**: A 3D visualization of a satellite in orbit around the Moon. The satellite is shown with its body axes (Body X, Body Y, Body Z) and a local coordinate system (WLH X, WLH Y, WLH Z). The background shows the Moon's surface.
- LogBook**: A table showing log entries with columns for Time, Grade, APS Name, Line, and Message. Examples include "CHECK TAOCMD4 RANGE 0.8824, 0.9024" and "CHECK success. Value = 0.892400026321411 [5474643F] [S:0]".
- LogBook**: A table showing log entries with columns for Time, Grade, APS Name, Line, and Message. Examples include "CHECK TAOCMD4 RANGE 0.8824, 0.9024" and "CHECK success. Value = 0.892400026321411 [5474643F] [S:0]".
- ITOS HostManager**, **Control Manager**, **Real Time Operation ...**, and **Telemetry View** are also visible at the bottom of the interface.