

COTS

“Commercial” is not always advertising...



Monica Alderighi

Astro-Siesta, 30/01/2014

COTS - Definition



- By Commercial Off-The-Shelf (COTS) is meant software or hardware products that are ready-made and available for sale or license
 - Manufacturer's standard products
 - Usually fast procurement
- Additional definition for COTS
 - Components with no prerequisite specifications with respect to the space environment (thermal, mechanical, radiation aspects)
 - Components at lower cost than similar rad-hard parts

Advantages of using COTS



- High computing performance, not available for space processors
- Large availability of support tools for SW development and tests
- Large availability of existing software libraries
- Compatibility with ground processors allows developing of low cost test environments and simulators

Concerns about using COTS

- Radiation effects
 - Radiation Tests results are often missing
- Variability
 - Process is likely to be modified at anytime, tracing the origin and manufacturing process is difficult
 - COTS devices has a variability from one manufacturer to another and for a single manufacturer
- Obsolescence
 - COTS suppliers constantly introduce new products, while
 - hardened OBC have a long development time and a long life cycle
- Reliability
 - Reliability data are often missing or incomplete
- No access to the internal design
 - Difficulty to fully characterize the design and to develop models

ESA initiative for using COTS in space

- **Project:** COTS based Computer for On Board systems (CoCs)
- **Objective:** Study and design on-board computing systems based on “Commercial Off-The-Shelf” components
- **Activity phase:**
 1. Design phase: defining the COTS computers as well as the methods for their manufacturing and qualification
 2. Implementation and qualification phase: manufacturing of breadboards that target real missions
- **3 H/W Contracts**
 - High Availability Computer - EADS-Astrium Germany
 - **High Reliability Computer - Thales Alenia Space Italia**
 - High Performance Computer - EADS-Astrium France

Hi Rel CoCs Project Team

Prime Contractor **ThalesAlenia**
A Thales / Finmeccanica Company *Space*

Sub-Contractors:

- project management and reporting,
- overall technical coordination
- interface with ESA and the Working Group
- Overall HiRel CoCs detailed specification
- FDIR strategy
- final technology trade-offs and selection
- definition of the CoCs evaluation methods and strategy.



Dept. of Automation and Computer Engineering of Politecnico di Torino (PoliTo):

- Survey of commercial off the shelf (COTS) processors
- Developing the CoC simulator
- Benchmark SW development

INAF



ISTITUTO NAZIONALE DI ASTROFISICA
NATIONAL INSTITUTE FOR ASTROPHYSICS

Institute IASF Milano/INAF

- Survey of Reprogrammable Logic Devices
- Hi-Rel CoCs Evaluation Environment & EGSE Definition
- EGSE Development



Department of Electronic Engineering & ULISSE Consortium of the University of Rome "Tor Vergata":

- Survey of candidate Memory Devices
- DDR-II ECC Development



SME company :

- Modeling of Hi-Rel CoCs Building Blocks
- Board and Basic SW Development

Hi Rel CoCs – Step 2 Objective

- Step 2 started in September 2011
- Activities are focusing on PM development and validation:
 - PM Board and FPGAs detailed design
 - Basic SW
 - PM Breadboard Manufacturing
 - **PM Board EGSE development**
 - **PM Breadboard Verification Test**
 - Development of benchmark Software
 - **PM Performances evaluation and Validation (including Faults injection)**
- Planned Step 2 activities completion by 4Q2014

PM Module - Major Requisites

- Outage duration in case of transient failure lower than 10 s
- Mean time between these outages higher than 30 days
- Targeted PM performance: 400 MIPS
- 3 high speed buses (200 Mb/s each), 3 low speed buses (1 Mb/s each), 100 low speed I/O (few kb/s each).
- Lifetime of 15 years
- Reliability better than 0.95 over 15 years

PM Module – Features



- CPU based on PPC 7448
- Working memory based on DDR-II
- Use High Speed FPGA (Virtex4) as Bridge
- Virtex4 scrubbing managed by external device
- Combination of SW and HW FDIR strategies
- HW Features specifically supporting SW FDIR
 - Selective Memory Protection
 - Individual Memory power switching to cope with SEFI
 - Smart watch-dog (supervisor) to check program flow
- ESA Standard data Interfaces
 - SpaceWire
 - High Speed Serial links

SBC PowerPC-7448 product definition

- It is the new TAS High Performance Processing Module, based on PowerPC 7448 (2300DMIPS@1GHz core clock), offering performances not available from other European Manufacturers.
- Development has been started in the frame of ESA COTS Based Computer and ARPA ASI Technology program.
- Space Qualified version development is going-on
- Envisageable Applications:
 - Optical Observation payloads
 - Radar Payload
 - Scientific Payloads
 - Planetary exploration Computers
 - Any application requiring high Processing performances



SBC PowerPC-7448

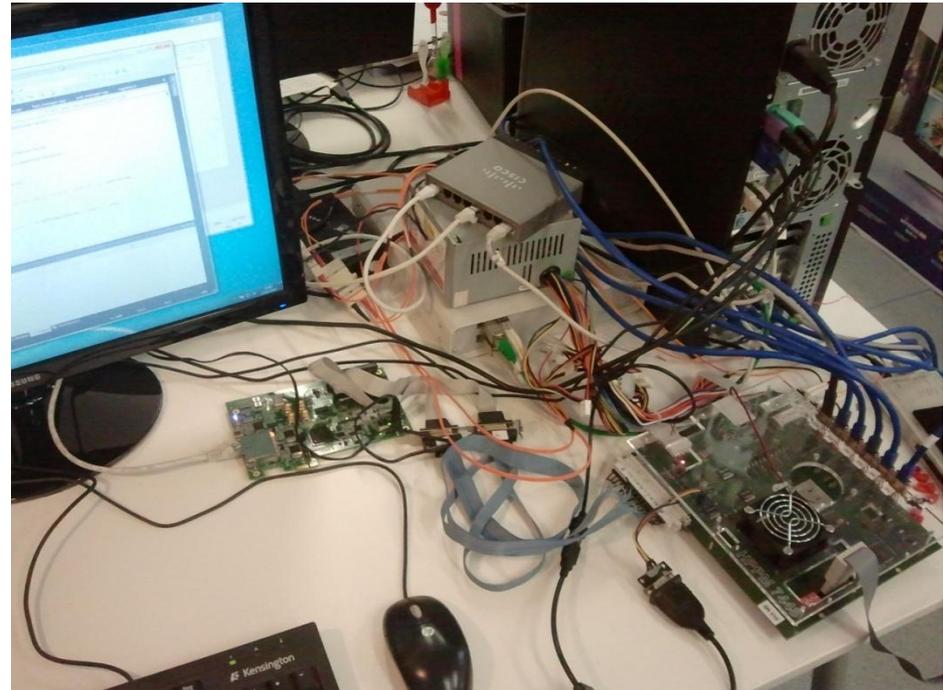
OPEN

© 2014, Thales Alenia Space

ThalesAlenia
A Thales / Finmeccanica Company
Space

EGSE HW/SW

- Powerful workstation PC, including
 - 6 SpaceWire ports
 - 1 Lauterbach Debugger tool
 - XILINX JTAG probe
 - ACTEL JTAG probe
 - 1 pci Digital I/O board
 - 2 Gigabit Ethernet boards
- Linux Ubuntu 10.4 OS
- C++ Programming Language
- Code Blocks 8.02 Development Environment



EGSE Functions



- PM Board testing/verification
 - allows testing specific resources of the PM board, such as memory, communications links, and data interfaces
- Benchmarking
 - allows testing PM board performances when selected benchmarks are applied
- Fault injection
 - allows testing the response of the PM board in presence of SEU like faults

EGSE Operation

Debugger environment

The screenshot displays the EGSE debugger environment. The top window is the 'Application Manager' showing the execution of a script for the 'egse_app' target. The middle window shows the 'Source code' of the 'main.cpp' file, which includes a main function with various variables and a loop. The bottom window shows the 'EGSE MENU' with options: 1 TEST MODE, 2 BENCHMARK MODE, 3 FAULT INJECTION MODE, and 4 EXIT. The user's selection is '1'. A red arrow points from the 'Debugger environment' title to the application manager window. Another red arrow points from the 'Source code' title to the source code window. A third red arrow points from the 'Application Manager run' title to the menu window.

Application Manager run

```
SCRIPT FOR DEBUG HIREL-BOARD
;init phase
cd /home/egse/pm_sw_prj/PM_EGSE_SW/
SYStem.CPU MPC7448
SYStem.Option.STEPSOFT ON
A:0:0004251C \dede\main\main
stopped at breakpoint
B::list
B::CD.PEDN /home/egse/pm_sw_prj/PM_EGSE_SW/laucmm
Save Save As... Save+Close Quit+Close Save+Do Do Debug
[ ]; SCRIPT FOR DEBUG HIREL-BOARD
;init phase
cd /home/egse/pm_sw_prj/PM_EGSE_SW/
SYStem.CPU MPC7448
SYStem.Option.STEPSOFT ON
A:0:00042528 [3b,0000] [11,0000] [r3,0x0]
egse_app
===== EGSE MENU =====
1 TEST MODE
2 BENCHMARK MODE
3 FAULT INJECTION MODE
4 EXIT
your selection -->1
```

Source code

```
int main (void)
{
int com, subscelta=0, result, num_car=1,tot_car=0, word=0,i, speed=0;
int test_code=1;
unsigned int address, length;
struct termios porta;
char buf[512], c;
char rx_string[512];
//FILE *datafile;
//char *fbuffer;
string fbuffer;
ifstream infile;
size_t flength;
int pos, byte_av, passo=0, esito;
char press;
bool yeah=false;
string spw_tx, spw_rx;
int startT,startR;
U32_receivedBytes;
unsigned char comm_buffer[6400];
fbuffer="";
length=fbuffer.size();
//for(i=0;i<50;i++)
// {
// data[i]=0;
// nomefile[i]=0;
// }
//strcat(nomefile, "/home/egse/work_dir/");
}
```

EGSE Fault Injection

- Evaluation of Fault PM module performance in presence of faults
 - Fault tolerance ability
 - Fault latency
 - Task duration
- Targets
 - Bridge FPGA
 - DDR II Volatile Memory
 - Non Volatile Flash Memory
 - PPC 7448 registers and L1/L2 memory
- Methods
 - Lauterbach Debugger tool
 - Device reconfiguration (Bridge FPGA)
 - Suitable software instrumentation



Thank you!!!!