A SOFTWARE ANALYSIS TOOL SUPPORTING FDIR MANAGEMENT FOR SYSTEMS WITH SPACE WIRE NETWORKS –MARC PROJECT

Session: Space Wire Missions and Applications

Short Paper

Omar Emam, Allan Whittaker, Simon Lentin, and Tony Jorden

EADS Astrium, Stevenage, Hertfordshire, SG1 2AS, UK

Wahida Gasti

ESA/ESTEC, 2200 AG Noordwijk ZH, The Netherlands

E-mail: omar.emam@astrium.eads.net , allan.whittaker@astrium.eads.net, simon.lentin@astrium.eads.net, tony.jorden@astrium.eads.net, wahida.gasti@esa.int

ABSTRACT

A custom-designed off-line software tool has been developed for analysing the Space Wire “SpW” network performance of a fault-tolerant system, based on the “Modular Architecture for Robust Computing”(MARC) concept, after the occurrence of a Fault Detection, Isolation and Recovery (FDIR) event. The tool also generates the FDIR and re-configuration tables required by the system’s onboard FDIR management software. This tool is referred to as the MARC ‘FDIR Analysis Tool’ which has been devised to support the design and implementation of MARC- based systems. The MARC concept is outlined, and the role of the “FDIR Analysis Tool” is described in this paper.

1 INTRODUCTION

The MARC project [1] aims at developing a fault-tolerant decentralised onboard computing architecture, using a high-reliability SpW network as its communication backbone. This is an UK-GSTP/ESA-funded project undertaken by Astrium UK, SciSys, and SEA. The FDIR Analysis tool has been developed by Astrium, as part of its role in specifying and developing the MARC FDIR strategy and related architecture. The tool will be used extensively in Astrium’s validation of the MARC concepts and designs, using the demonstrator hardware [2] and software, towards the end of 2010.

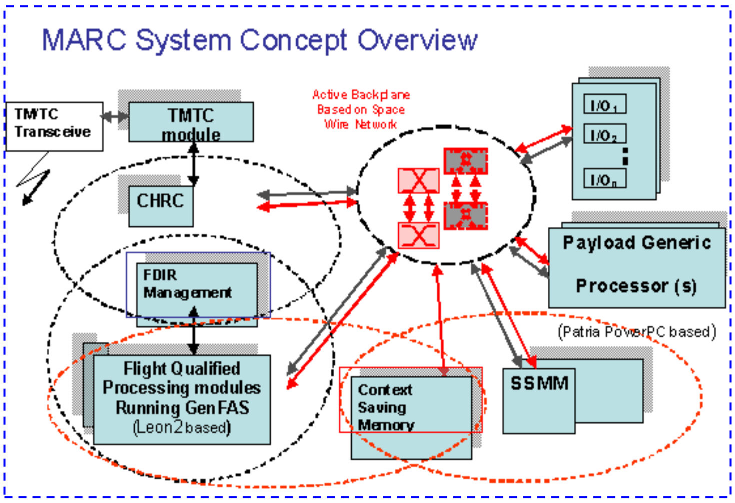
The complexity of advanced networking systems, with their multitude of parameters to be taken into account, makes it almost impossible to design an optimum network solution simply by manual inspection. The use of some form of system analysis tool has therefore become essential to support the design of complex computing networks where there are multiple data exchange paths, with different data traffic characteristics and constraints.

MARC is one such advanced computing network system for which the MARC ‘FDIR analysis tool’ was conceived, as an off-line software application that would be run at the design phase of this complex system. The tool is used to analyse the suitability of a given MARC SpW network, in terms of its throughput and latency, to meet the system requirements. The tool is also used to enable the user to run FDIR scenarios which are intended to analyse the system performance after a fault recovery and highlight any non-conformances. The tool then generates the FDIR and re-configuration tables associated with these FDIR conditions. The resulting tables are used by Astrium's MARC 'FDIR Manager' [4] which is implemented by SciSys as part of their Generic Fault-tolerant Software Architecture "GenFAS" software framework [1] developed for MARC.

2 FDIR ANALYSIS TOOL AS PART OF MARC

In a MARC system multiple computing, memory, and Input/output (I/O) nodes (or modules) are interconnected via a high-reliability SpW network. MARC is a scalable architecture. In its simplest form it resembles a standard on-board computing system, with a set of prime and redundant nodes/modules, all connected to single pair of prime and redundant SpW routers to form a simple single-hop (single router) network. In this case it is relatively simple to ensure, by inspection, that the resulting network will meet the specified performance requirements. On the other extreme, up to 24 nodes of different types (computing, memory, and I/Os) can be connected to a number of chained routers to form a complex heterogeneous network. The problem of designing the system is made more difficult since it is possible to have multiple data paths between nodes with different packet lengths as well as different data rates and link speeds. In addition, there may be requirements on data traffic prioritisation, in that some data packets, such as command, Health-Status or Timing-Data packets have to be routed from source to destination within a restricted timing window. In this case, designing a system that guarantees the specified performance cannot be achieved without the aid of some analysis tool. Finally, the tool has to enable the user to generate the FDIR and re-configuration tables required by "FDIR manager" onboard software which is used to implement the FDIR strategy. No off-the-shelf tools have been identified which meets these specific needs of a MARC system. In particular, a network analysis tool for MARC must include the capability to define the parameters for the exchange of the Health Status messages necessary for implementing the MARC FDIR strategy.

3 HOW THE TOOL WORKS

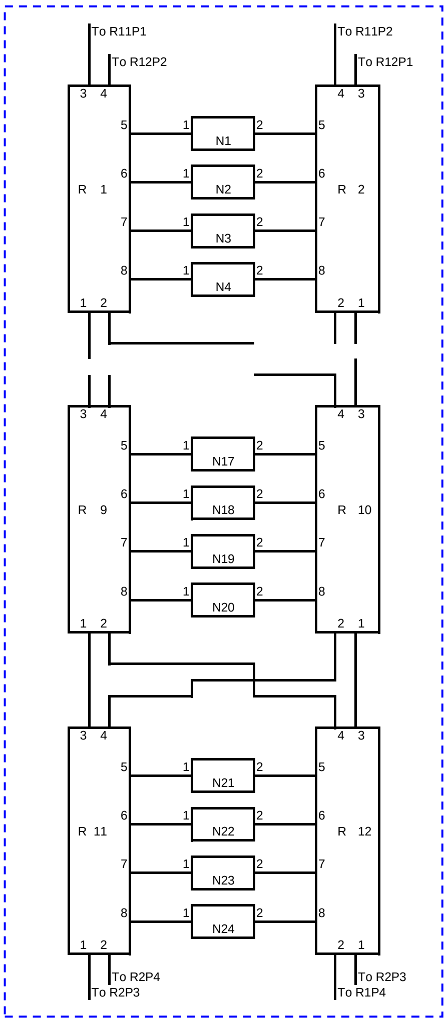

MARC System Concept Overview
TM/TC Transceive
TMTC module
Active Backplane Based on Space Wire Network
I/O1
I/O2
I/On
CHRC
Payload Generic Processor (s)
(Patria PowerPC based)
FDIR Management
Flight Qualified Processing modules Running GenFAS (Leon2 based)
Context Saving Memory
SSMM

The tool takes as its input a user defined file that specifies the SpW network architecture being considered for analysis. The first stage of this analysis is to determine what all possible network connections exist between the various nodes. The tool also checks that the network conforms to the MARC standard network architecture - see following figure, as this is important for generating the FDIR related tables. Once the tool has established that any node can connect to any other node via at least two paths, the network performance analysis can then start. This analysis is used to validate the robustness of the MARC network in terms of fault tolerance, by enabling the user to mimic failure and failure recovery in the system and identify any non-compliances in its performance after recovery. For example swapping a prime computing node on one router with a redundant node on another router and showing that the resulting network still meets the system performance requirements.

By performing the validation for all possible network failures and failure recovery conditions the tool is able to generate a set of FDIR and configuration tables that the on-board FDIR Manager software uses to indentify a fault recovery procedure. The onboard software combines the entry in the table, pointing to the current configuration, with the identifier of the diagnosed fault to generate a pointer. This allows it to identify an entry in the FDIR and configuration table that defines the configuration of the recovered system.

3.1 SPACE WIRE NETWORK PERFORMANCE ANALYSIS

The network performance analysis has two main objectives: to analyse the throughput of the network and to determine the latency of data packets passing through it. In the analysis, the tool allows for packets of different sizes and different rates, as well as two different Space Wire link frequencies (see below). In addition, the tool permits multiple data communication channels between any node and any other node, which creates a 'virtual network'. This is important since it enables the user to allocate the different data channels different priorities for a SpW-VN [5] based system, or different time slots for a SpW-RT based system [3].



In summary, the key steps for network performance analysis are:

Throughput:

- Determine which of multiple possible paths are used for a node-node connection.
- Calculate and display the percentage loading contribution from all data communication channels in a link.
- Flag if the total loading percentage of a link exceeds a defined threshold.

Latency:

- Calculate Latency for a particular path (sum of router & link contributions).
- Routers –calculate a packet's port-to-port time: worst case, all other ports are already occupied.
- Latency is assessed for each data communication channel (of a virtual network, if there are multiple paths per link).

Both the latency and the throughput takes into account that the network can be heterogeneous in terms of SpW link frequencies, being either 100MHz or 10MHz.

Latency Algorithm:
(The following values are based on using Atmel 10X SpW router):

T_{sl}	switching latency:	a constant value of 133 nanoseconds,
T_{ppl}	port to port latency:	a constant value of 547 nanoseconds
T_{bpb}	bits per byte:	a constant value of 8 bits.
T_{bpp}	bytes per packet:	32 – 1M bytes (typically 1024)
T_{bps}	bits per second	in the range of 80-70% of SpW link frequency.
T_{ppr}	ports per router:	8 ports

T_{fbd}	first byte delay:	$T_{sl} + T_{ppl}$
T_{spb}	seconds per byte:	T_{bpb} / T_{bps}
T_{spp}	seconds per packet:	$T_{spb} \times T_{bpp}$
T_{tropp}	time router occupied per port:	$T_{spp} + T_{fbd}$
T_{mtro}	max time router occupied:	$T_{tropp} \times (T_{ppr} - 2)$

3.2 MARC FDIR AND CONFIGURATION TABLES GENERATION

The operator starts the FDIR and configuration tables' generation by selecting the Generate Tables function in the Traffic Results Window. The system performs the following sequence of actions to generate FDIR tables for the pre-defined logical network:

1. Generate a bit pattern in a 64-bit word to represent the defined system 'start' configuration word, (i.e. assuming no failures), where a 1 indicates an active component and 0 represents an inactive component. When the flight system is in the corresponding configuration, the FDIR manager uses this as an "**index"** to the FDIR data it needs. For that configuration word, the router tables are then written into the Configuration Table.

2. For that configuration an element to be simulated as the 'failed' element is selected. The tool works out the new set of active elements, given the presumed failure and the known redundancies. The tool then analyses the new configuration to see whether it meets throughput & latency constraints. If not, it will write out a special configuration word defining a "safe mode" to be adopted when this faulty configuration is reached and sets a bit representing a 'non-operational flag'.

3. If the new configuration **does** meet throughput and latency constraints, the tool will write the configuration to the FDIR and configuration Tables. The sequence repeats from step 2, but with a new element chosen to be the 'failed' element. This continues until all elements in turn have taken the role of the 'failed' element.

4. When all elements for that initial start up configuration have been exhausted then a new 'start' configuration is defined, and the whole sequence is restarted from step 1. The overall process finishes when all combination of failures have been exhausted for all the start configurations. The failures should be cumulative, such that, for example, a 12 node system results in 2^{12} configurations.

4 WHAT THE TOOL LOOKS LIKE

The tool expects the system architecture to conform to that agreed within the MARC framework in that it consists of a number of different types of nodes which are interconnected via SpW links. The nodes may be processor platforms, memory modules, I/O modules, routers, etc. The user specifies the number and type of nodes as well as the interconnect between them. The tool includes a basic model of each type of node. The analysis tool has the following key user interaction stages:

4.1 THE CONFIGURATION INPUT FILE

This represents a file of network configuration data which is created before running the analysis tool. It contains text which defines what each port of each element in the physical network, is connected to. The format is:

<Router>,<Port>,<Destination>,<Link_Speed>

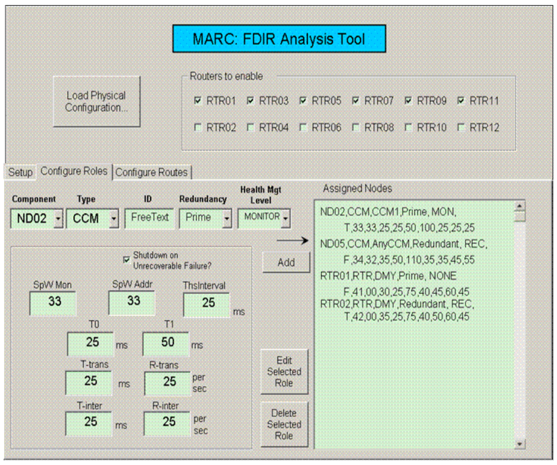
For example:
RTR01,1,RTR03,100
Meaning: Router 1, port 1 is connected to Router 3 and has a link speed of 100 MHz
ND01,1,RTR01,10
Meaning: Node 1, port 1 is connected to Router 1 and has a link speed of 10 MHz

4.2 THE PHYSICAL NETWORK WINDOW

This is the first stage in the network analysis process. Here the operator loads the "Configuration Input file" which the analysis tool then examines by performing some context checking to ensure that there were no obvious errors in the configuration file. From the GUI, the user can manually select or modify the configuration of the prime and redundant routers in the network. The analysis tool then determines the number of links (hop-numbers) between any two nodes on the network and displays it as a matrix. The user can then select an individual cell of the matrix to get information on the network links associated with that node-to-node data path.

4.3 LOGICAL NETWORK WINDOWS

This window allows the operator to assign roles (functions) to each of the nodes in the physical network, including whether the role is prime or redundant. The operator is also given the facility to define the routes between prime nodes and what level of data traffic will be carried.



Routers to enable
[✓ RTR01] [✓ RTR03] [✓ RTR05] [✓ RTR07] [✓ RTR09] [✓ RTR11]
RTR02 RTR04 RTR06 RTR08 RTR10 RTR12

Component	Type	ID	Redundancy	Health Mgt Level
ND02	CCM	FreeText	Prime	MONITOR

[✓ Shutdown on Unrecoverable Failure?]

SpW Mon: 33
SpW Addr: 33
ThsInterval: 25 ms
T0: 25 ms
T1: 50 ms
T-trans: 25 ms
R-trans: 25 per sec
T-inter: 25 ms
R-inter: 25 per sec

Assigned Nodes
```
ND02,CCM,CCM1,Prime, MON,
    T,33,33,25,25,50,100,25,25,25
ND05,CCM,AnyCCM,Redundant, REC,
    F,34,32,35,50,110,35,35,45,55
RTR01,RTR,DMY,Prime, NONE
    F,41,00,30,25,75,40,45,60,45
RTR02,RTR,DMY,Redundant, REC,
    T,42,00,35,25,75,40,50,60,45
```

4.4 TRAFFIC RESULTS WINDOW

This window allows the operator to initiate analysis of the pre-defined logical network. It is from this window that the FDIR and configuration tables' generation can be initiated.

5 CONCLUSION

A complex SpW network based computing system could not realistically be designed and implemented without the aid of some analysis tool. In the case of MARC the 'FDIR analysis tool' is an essential part of the system and is key to generating the FDIR and configuration tables that lie at the heart of the MARC FDIR strategy.

6 REFERENCES

1. Gasti et al, "Modular Architecture for Robust Computation – MARC", ISC2008 Conference. Nara-Japan, Nov 2008

2. Senior et al, "Modular Architecture for Robust Computation – MARC", ISC2007 Conference, Dundee-UK, Sept 2007

3. Fowell et al, "the Adaptation and Implementation of Space Wire-RT for the MARC Project", ISC2010 Conference, St.Petersburg-Russia , June 2010

4. Emam et al, "An FDIR Strategy based on Message Exchange Approach to Implement Autonomous FDIR Management on the MARC System", DASIA 2010 Conference, Budapest- Hungary, June 2010

5. Barry et al, "Virtual SpaceWire Networks: A Mechanism for Real-Time Applications", SpaceWire Working Group Meeting 13, ESTEC - Noordwijk - The Netherlands, Sept 2009