COUPLING TOOLS FOR SPACEWIRE ON-BOARD NETWORK: SIMULATION, CONFIGURATION AND VALIDATION

ISC2022: Paper number 71, B.Attanasio

Date: 17/10/2022

Ref: xxxxx

Template: 83230347-DOC-TAS-EN-006

TOOLS PRESENTATION

/// **MOST** for Modelling of Spacecraft Traffic (not only SpW, but SpF, 1553, CAN)

/// Goal of MOST:

- To support SpW network design and optimization
- To allow SpW networks performances analysis from the beginning, without waiting for system testing phase
- To offer a progressive tool for SpW experts who would like to integrate specific SpW components, or to update existing library with regard to standard upgrades

/// MOST simulator is dedicated to the following users:

- System engineers who have to design network topology and to perform validation tests.
- Developers who would need to test new component features or new protocol.

Does MOST behave like a real network?

/// **SPACEMAN** is a SpaceWire Network Management Tool used to discover and configure a network using some features of either the NDCP protocol or the RMAP protocol.

/// It sends request over the Network and discover the nodes depending on the answers.

RMAP REGISTERS IMPLEMENTATION IN MOST SIMULATOR

RMAP NODE IMPLEMENTATION

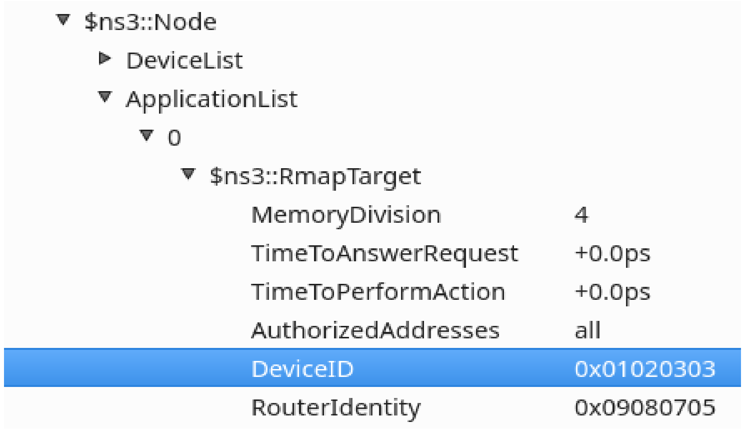
RMAP logic implemented/modified specifically at node level:

/// The implemented registers are:

- The DeviceID register (0x105): Identity number
- The Router Identity register (0x101): Identity number
- The Network Discovery Register (0x100): Dynamic Status Information
- The general purpose register (0x106): Special Memory to Write and then Read

/// Real computer Memory reserved and managed at C++ level

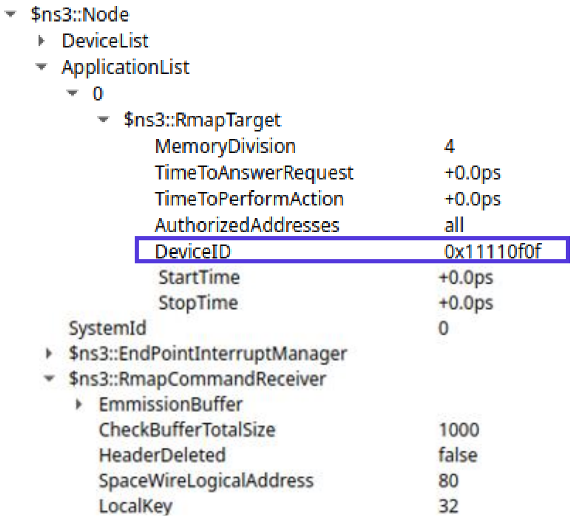
- Read and Write in the corresponding memories



```
▼ $ns3::Node
    ▶ DeviceList
    ▼ ApplicationList
        ▼ 0
            ▼ $ns3::RmapTarget
                  MemoryDivision          4
                  TimeToAnswerRequest     +0.0ps
                  TimeToPerformAction     +0.0ps
                  AuthorizedAddresses     all
                  DeviceID                0x01020303
                  RouterIdentity          0x09080705
```

RMAP NODE ATTRIBUTES

Configuration of a Device ID



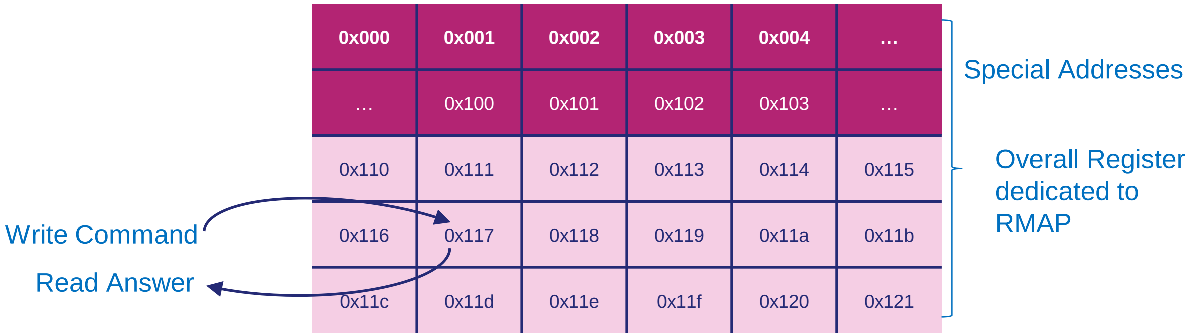
```
▾ $ns3::Node
   ▸ DeviceList
   ▾ ApplicationList
      ▾ 0
         ▾ $ns3::RmapTarget
              MemoryDivision              4
              TimeToAnswerRequest         +0.0ps
              TimeToPerformAction         +0.0ps
              AuthorizedAddresses         all
              DeviceID                    0x11110f0f
              StartTime                   +0.0ps
              StopTime                    +0.0ps
        SystemId                          0
   ▸ $ns3::EndPointInterruptManager
   ▾ $ns3::RmapCommandReceiver
      ▸ EmmissionBuffer
        CheckBufferTotalSize              1000
        HeaderDeleted                     false
        SpaceWireLogicalAddress           80
        LocalKey                          32
```

New Registers implemented

Real computer Memory reserved

A memory that contains registers has been implemented in the RMAP node.

An RMAP query can access any register in memory, ranging from address 0x0 to address 0x200 (512).



0x000	**0x001**	**0x002**	**0x003**	**0x004**	**…**
…	0x100	0x101	0x102	0x103	…
0x110	0x111	0x112	0x113	0x114	0x115
0x116	0x117	0x118	0x119	0x11a	0x11b
0x11c	0x11d	0x11e	0x11f	0x120	0x121

NETWORK DISCOVERY REGISTER FOR RMAP NODE: EXPLANATION

/// No configuration in the GUI: Automatically managed by the node

/// Structure of the Network Discovery Register for a node

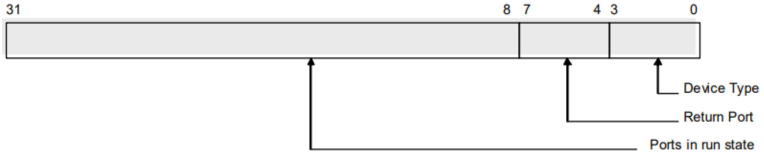


Figure 9-4 Network Discovery Register Fields

/// For a device the values are the following ones:

- Not a Router so considered as “Unknown Device”: 0000
- Only 1 port so always the same return port
- Only one port and always active so run state is constant and equal to 1

Bits 3:0 : 0000 (device Type)
Bits 7:4 : 0001 (return port = port 1)
Bits 31:8 : 0…000000001 (port 1 in Run State)

/// Also done for the router shown later in the validation part.

RMAP INTEGRATION IN 10X ROUTER

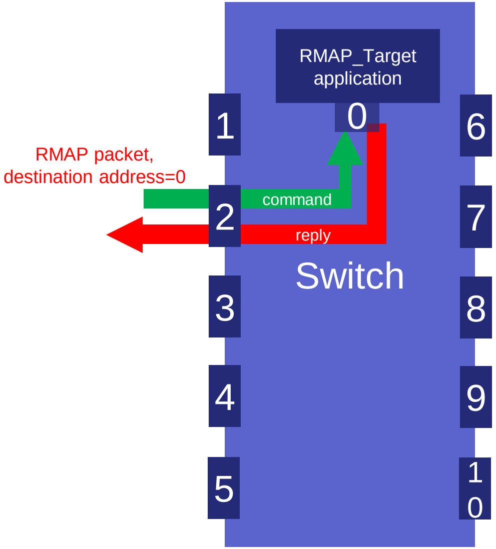
Modification of the Router Structure

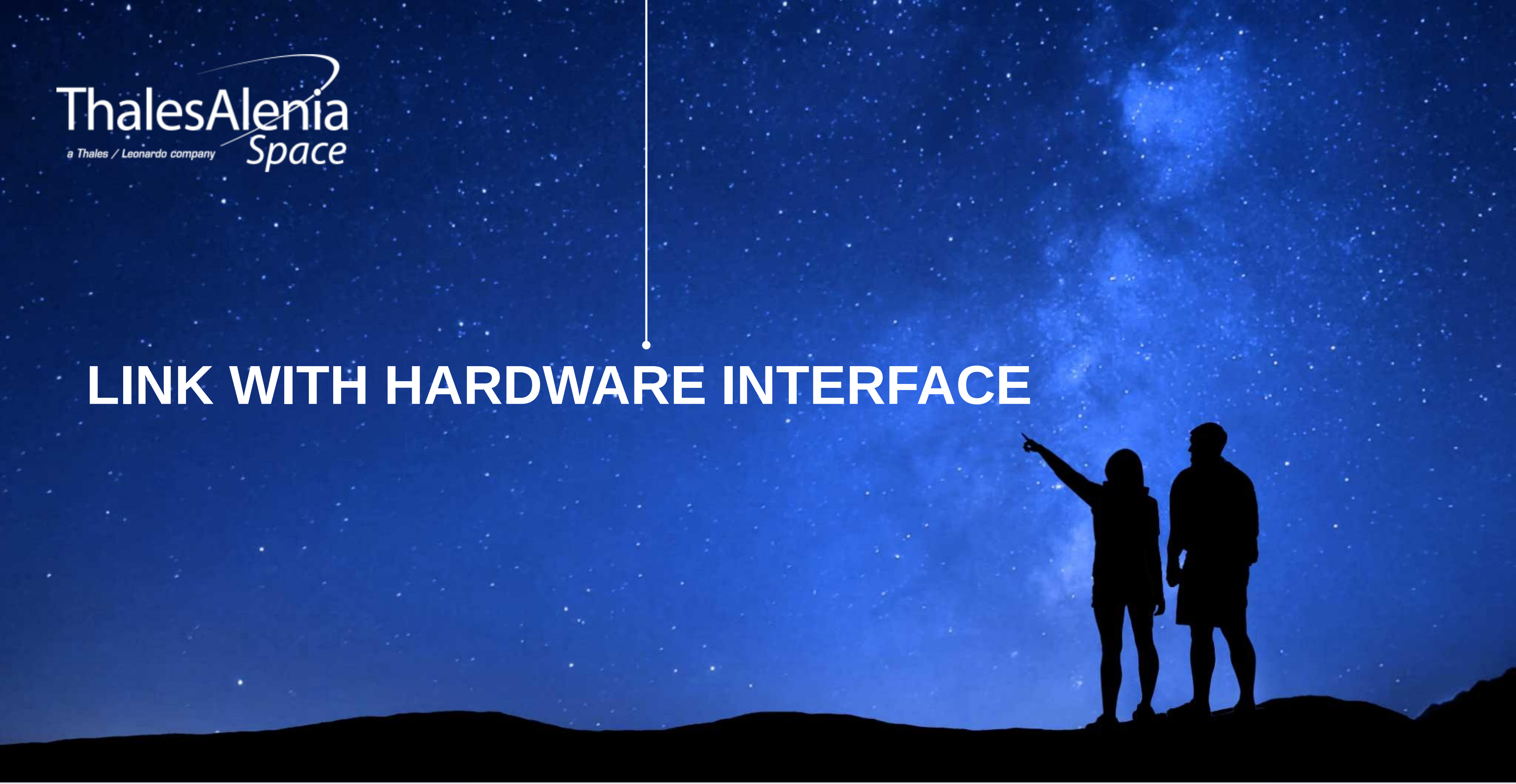
Port 0 is a Logical Port and is fully internal to the switch

The RMAP application produces a reply which is subsequently sent back through the port where the initial RMAP command arrived through.

Packets sent to any port of the router with a target address 0 are routed to the RMAP application of the router (internally without any real HW port).

The Device ID register and the Router identity register are accessed exactly the same way as the Device ID register of the RMAP node.





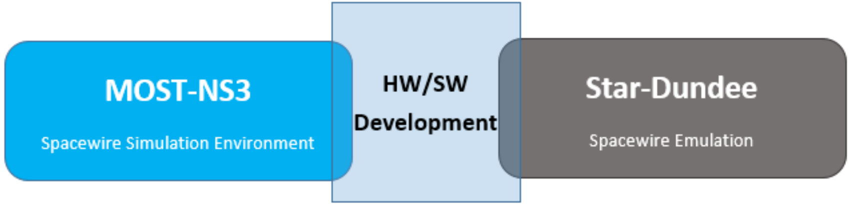
LINK WITH HARDWARE INTERFACE

IMPLEMENTATION HW/SW NODE IN MOST

Main Principle:

/// Add a new MOST Building Block

- Goal is to make the bridge between the simulation world and the Real HW World
- Use the STAR-System libraries to interface with the board



STAR-DUNDEE SpaceWire PCI Express board:

/// Connected to the development PC for MOST

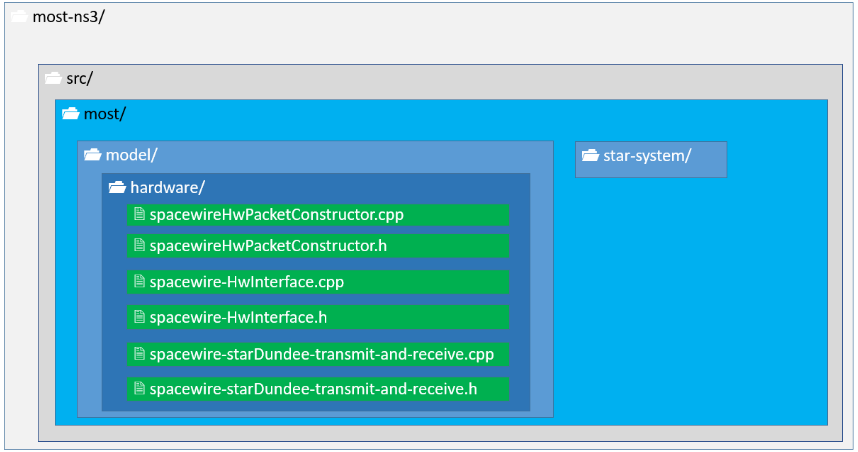
- Provides 3 SpaceWire interfaces
- And a PCI express connection

FILES ARCHITECTURE & NODE BUILDING

Code Architecture:

/// New Folder in "*model*" representing a new MOST building block

/// Star-System is a folder apart where the libraries for the board are implemented

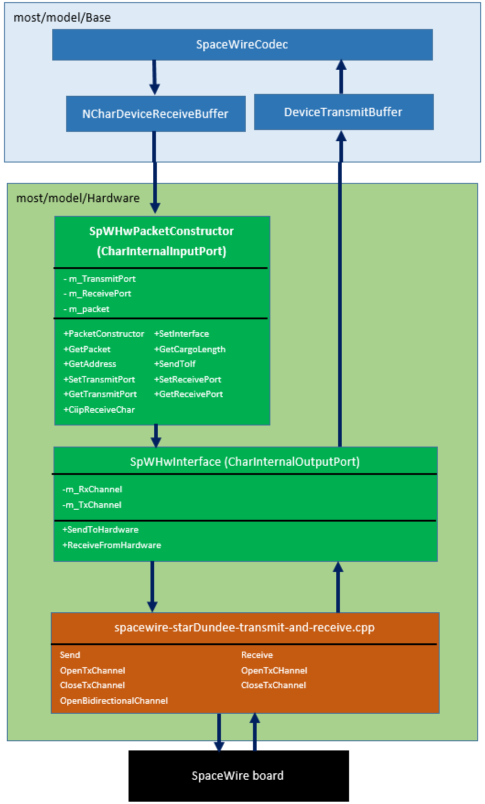


most-ns3/
src/
most/
model/
hardware/
spacewireHwPacketConstructor.cpp
spacewireHwPacketConstructor.h
spacewire-HwInterface.cpp
spacewire-HwInterface.h
spacewire-starDundee-transmit-and-receive.cpp
spacewire-starDundee-transmit-and-receive.h
star-system/

Blue: Existing Spacewire Basis functions

Green: functions developed for HW/SW Node

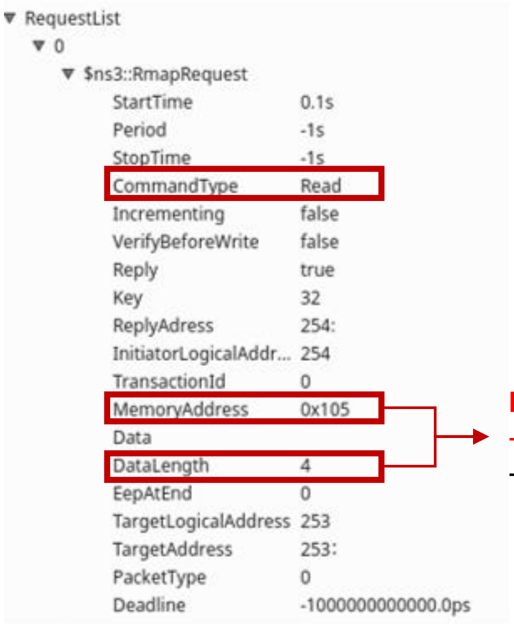
Orange: star-system functions used (not all) ➔ Board dependant



INTER FUNCTIONING WITH SPACEMAN

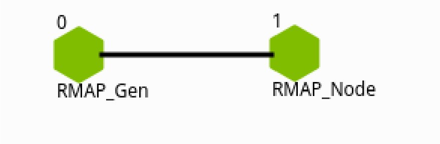
VALIDATION AT MOST LEVEL : NODE – 1/4

Traffic definition in MOSTGui at the emitter level: Request with the Target Memory and Data Length for an Rmap Node



- RequestList
  - 0
    - $ns3::RmapRequest

StartTime	0.1s
Period	-1s
StopTime	-1s
CommandType	Read
Incrementing	false
VerifyBeforeWrite	false
Reply	true
Key	32
ReplyAdress	254:
InitiatorLogicalAddr...	254
TransactionId	0
MemoryAddress	0x105
Data	
DataLength	4
EepAtEnd	0
TargetLogicalAddress	253
TargetAddress	253:
PacketType	0
Deadline	-1000000000000.0ps

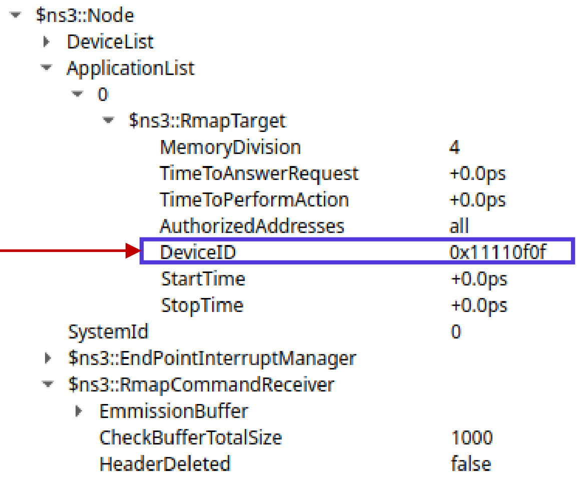


Read command

DeviceID register:

- **at address 0x105**
- 32 bits (4 bytes)

Node Configuration in MOSTGui at the receiver level



- $ns3::Node
  - DeviceList
  - ApplicationList
    - 0
      - $ns3::RmapTarget

MemoryDivision	4
TimeToAnswerRequest	+0.0ps
TimeToPerformAction	+0.0ps
AuthorizedAddresses	all
DeviceID	0x11110f0f
StartTime	+0.0ps
StopTime	+0.0ps

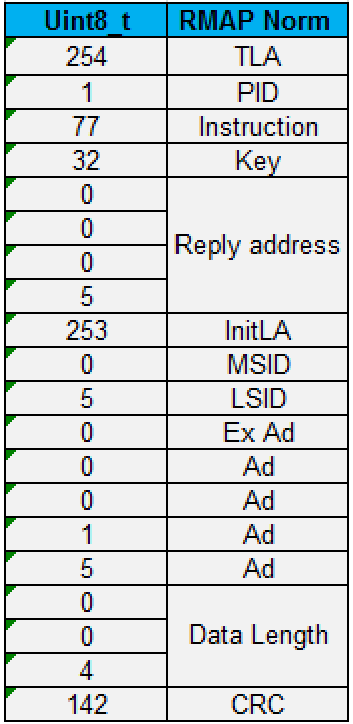
SystemId	0

- $ns3::EndPointInterruptManager
- $ns3::RmapCommandReceiver
  - EmissionBuffer

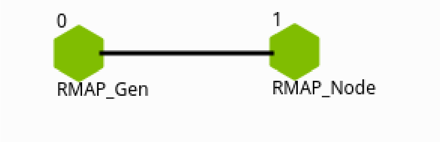
CheckBufferTotalSize	1000
HeaderDeleted	false

VALIDATION AT MOST LEVEL : NODE – 2/4

Command sent by SPACEMAN

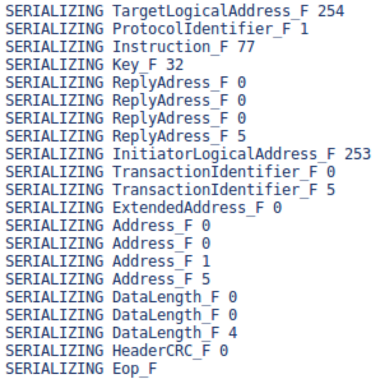


Uint8_t	RMAP Norm
254	TLA
1	PID
77	Instruction
32	Key
0	Reply address
0	
0	
5	
253	InitLA
0	MSID
5	LSID
0	Ex Ad
0	Ad
0	Ad
1	Ad
5	Ad
0	Data Length
0	
4	
142	CRC



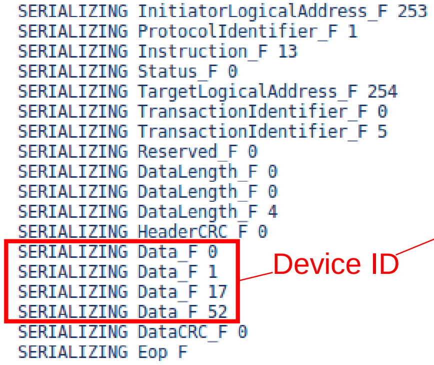
Degenerated test, not used with SPACEMAN because only one node and not a full network.

Command sent by the generator in MOST



```
SERIALIZING TargetLogicalAddress_F 254
SERIALIZING ProtocolIdentifier_F 1
SERIALIZING Instruction_F 77
SERIALIZING Key_F 32
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 5
SERIALIZING InitiatorLogicalAddress_F 253
SERIALIZING TransactionIdentifier_F 0
SERIALIZING TransactionIdentifier_F 5
SERIALIZING ExtendedAddress_F 0
SERIALIZING Address_F 0
SERIALIZING Address_F 0
SERIALIZING Address_F 1
SERIALIZING Address_F 5
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 4
SERIALIZING HeaderCRC_F 0
SERIALIZING Eop_F
```

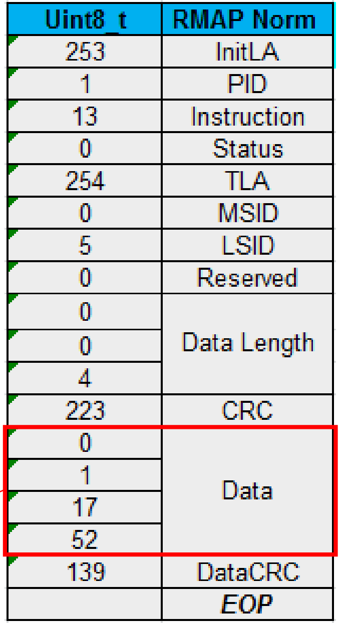
Reply sent by the target in MOST



```
SERIALIZING InitiatorLogicalAddress_F 253
SERIALIZING ProtocolIdentifier_F 1
SERIALIZING Instruction_F 13
SERIALIZING Status_F 0
SERIALIZING TargetLogicalAddress_F 254
SERIALIZING TransactionIdentifier_F 0
SERIALIZING TransactionIdentifier_F 5
SERIALIZING Reserved_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 4
SERIALIZING HeaderCRC_F 0
SERIALIZING Data_F 0
SERIALIZING Data_F 1
SERIALIZING Data_F 17
SERIALIZING Data_F 52
SERIALIZING DataCRC_F 0
SERIALIZING Eop_F
```

Device ID

Reply received by SPACEMAN

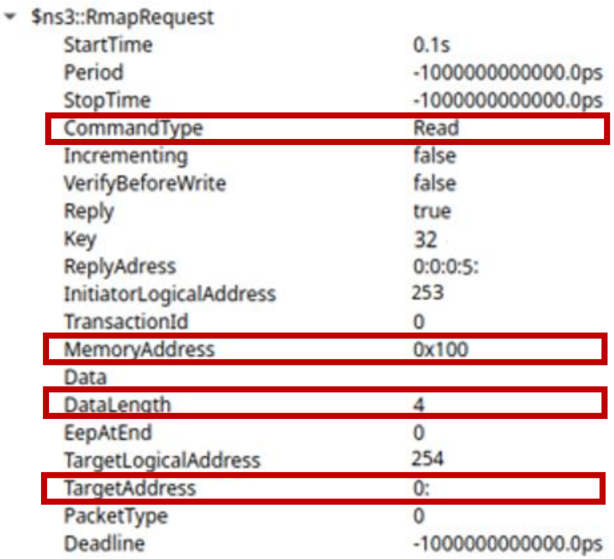


Uint8_t	RMAP Norm
253	InitLA
1	PID
13	Instruction
0	Status
254	TLA
0	MSID
5	LSID
0	Reserved
0	Data Length
0	
4	
223	CRC
0	Data
1	
17	
52	
139	DataCRC
	EOP

VALIDATION AT MOST LEVEL : ROUTER – 3/4

Network Discovery Register

Traffic definition in MOSTGui at the emitter level: Request with the Target Memory and DataLength



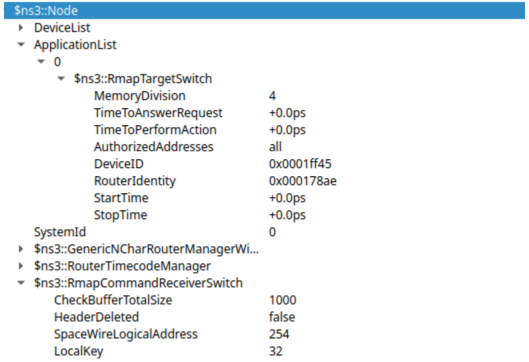
```
▾ $ns3::RmapRequest
    StartTime                 0.1s
    Period                    -1000000000000.0ps
    StopTime                  -1000000000000.0ps
    CommandType               Read
    Incrementing              false
    VerifyBeforeWrite         false
    Reply                     true
    Key                       32
    ReplyAdress               0:0:0:5:
    InitiatorLogicalAddress   253
    TransactionId             0
    MemoryAddress             0x100
    Data
    DataLength                4
    EepAtEnd                  0
    TargetLogicalAddress      254
    TargetAddress             0:
    PacketType                0
    Deadline                  -1000000000000.0ps
```

Network discovery register:

- **at address 0x100**
- 32 bits (4 bytes)

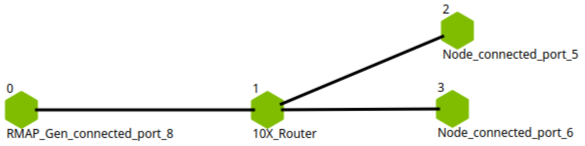
Destination address 0 to reach the configuration port

Node Configuration in MOSTGui at the 10X router



```
$ns3::Node
  ▸ DeviceList
  ▾ ApplicationList
      ▾ 0
          ▾ $ns3::RmapTargetSwitch
              MemoryDivision              4
              TimeToAnswerRequest         +0.0ps
              TimeToPerformAction         +0.0ps
              AuthorizedAddresses         all
              DeviceID                    0x0001ff45
              RouterIdentity              0x000178ae
              StartTime                   +0.0ps
              StopTime                    +0.0ps
    SystemId                              0
  ▸ $ns3::GenericNCharRouterManagerWi...
  ▸ $ns3::RouterTimecodeManager
  ▾ $ns3::RmapCommandReceiverSwitch
        CheckBufferTotalSize              1000
        HeaderDeleted                     false
        SpaceWireLogicalAddress           254
        LocalKey                          32
```

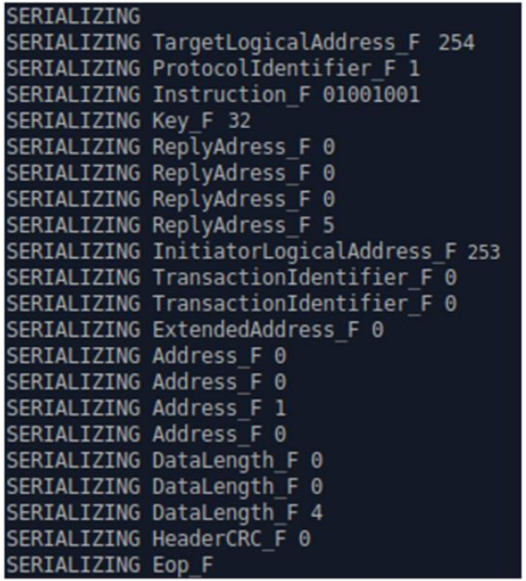
Topology: only ports 5, 6 and 8 will be in the RUN STATE



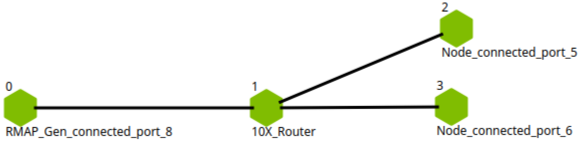
VALIDATION AT MOST LEVEL : ROUTER – 4/4

Network Discovery Register

Command sent by SPACEMAN



```
SERIALIZING
SERIALIZING TargetLogicalAddress_F  254
SERIALIZING ProtocolIdentifier_F 1
SERIALIZING Instruction_F 01001001
SERIALIZING Key_F 32
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 0
SERIALIZING ReplyAdress_F 5
SERIALIZING InitiatorLogicalAddress_F 253
SERIALIZING TransactionIdentifier_F 0
SERIALIZING TransactionIdentifier_F 0
SERIALIZING ExtendedAddress_F 0
SERIALIZING Address_F 0
SERIALIZING Address_F 0
SERIALIZING Address_F 1
SERIALIZING Address_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 4
SERIALIZING HeaderCRC_F 0
SERIALIZING Eop_F
```



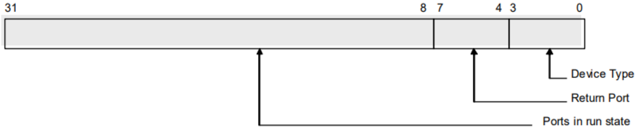
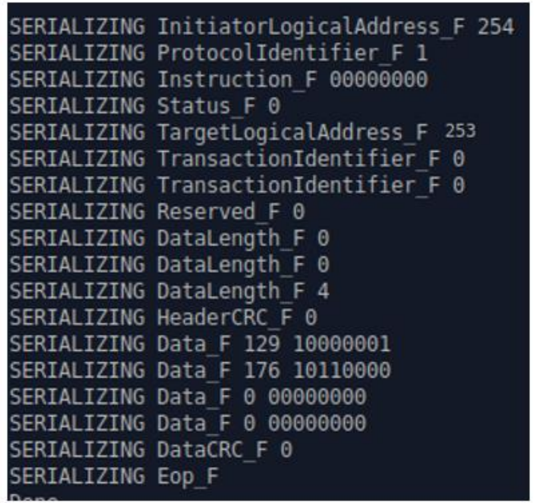


Figure 9-4 Network Discovery Register Fields

Bits 3:0 : 0001 (router)
Bits 7:4 : 1000 (return port = port 8)
Bits 31:8 : 0…010110000 (port 5,6, and 8 in run state)
…987654321

Automatic update of the return port at every RMAP Request

Reply received by SPACEMAN

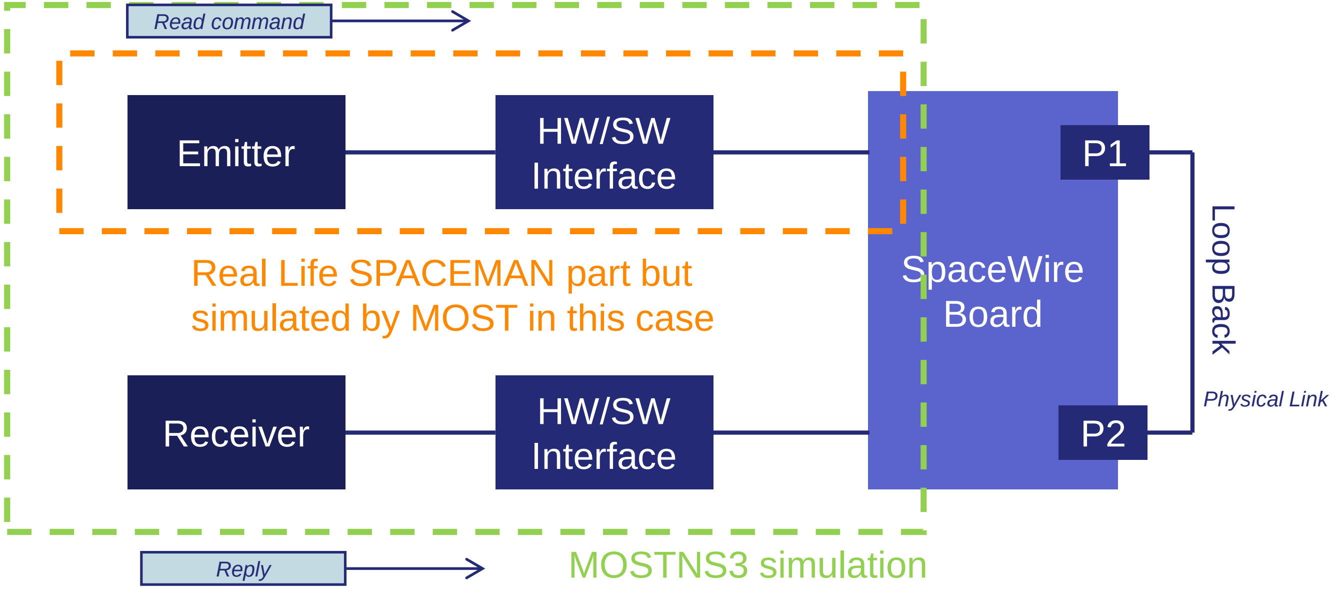


```
SERIALIZING InitiatorLogicalAddress_F 254
SERIALIZING ProtocolIdentifier_F 1
SERIALIZING Instruction_F 00000000
SERIALIZING Status_F 0
SERIALIZING TargetLogicalAddress_F 253
SERIALIZING TransactionIdentifier_F 0
SERIALIZING TransactionIdentifier_F 0
SERIALIZING Reserved_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 0
SERIALIZING DataLength_F 4
SERIALIZING HeaderCRC_F 0
SERIALIZING Data_F 129 10000001
SERIALIZING Data_F 176 10110000
SERIALIZING Data_F 0 00000000
SERIALIZING Data_F 0 00000000
SERIALIZING DataCRC_F 0
SERIALIZING Eop_F
Done
```

RMAP	C	Rd	253	254	27	32		0x0000000100	4	20	22	00 00 00 05		01 00	fe 01 4d 20	00 00 00 05	fd 00 1b 00	00 00 01 00	00 00 04 bb	***EOP***
RMAP	R	Rd	254	253	27		OK		4	17	17				fd 01 0d 00	fe 00 1b 00	00 00 04 3e	00 0f 01 10	47 ***EOP***	

PROPRIETARY INFORMATION
This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales Alenia Space.

VALIDATION AT MOST LEVEL WITH SPACEMAN BEHAVIOR – 1/3

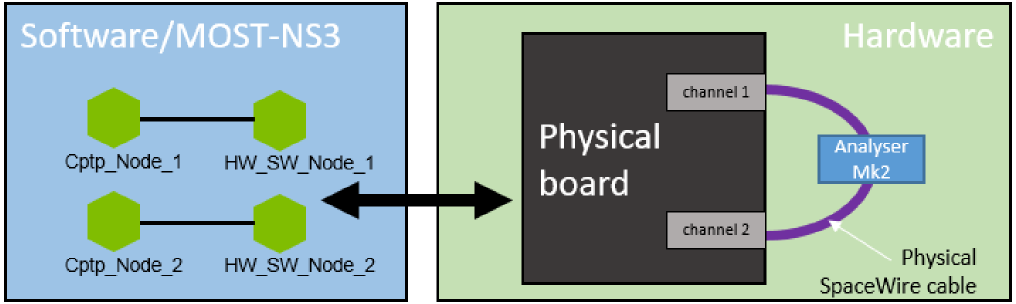


This simulation was done entirely on MOSTGui: RMAP_HW_SW_LOOPBACK

FINAL END-TO-END TESTS

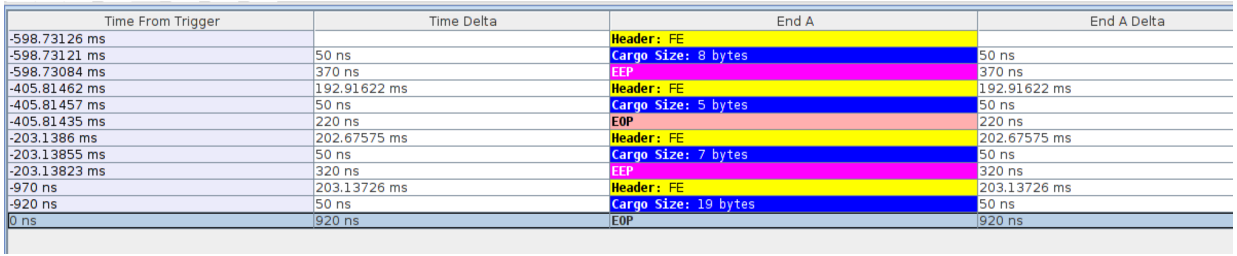
Loopback Tests:

- Emitter part with CPTP Node + HW/SW Node
- Receiving part with CPTP Node + HW/SW Node



Sniffer:

- Check on the physical link that packets are well sent

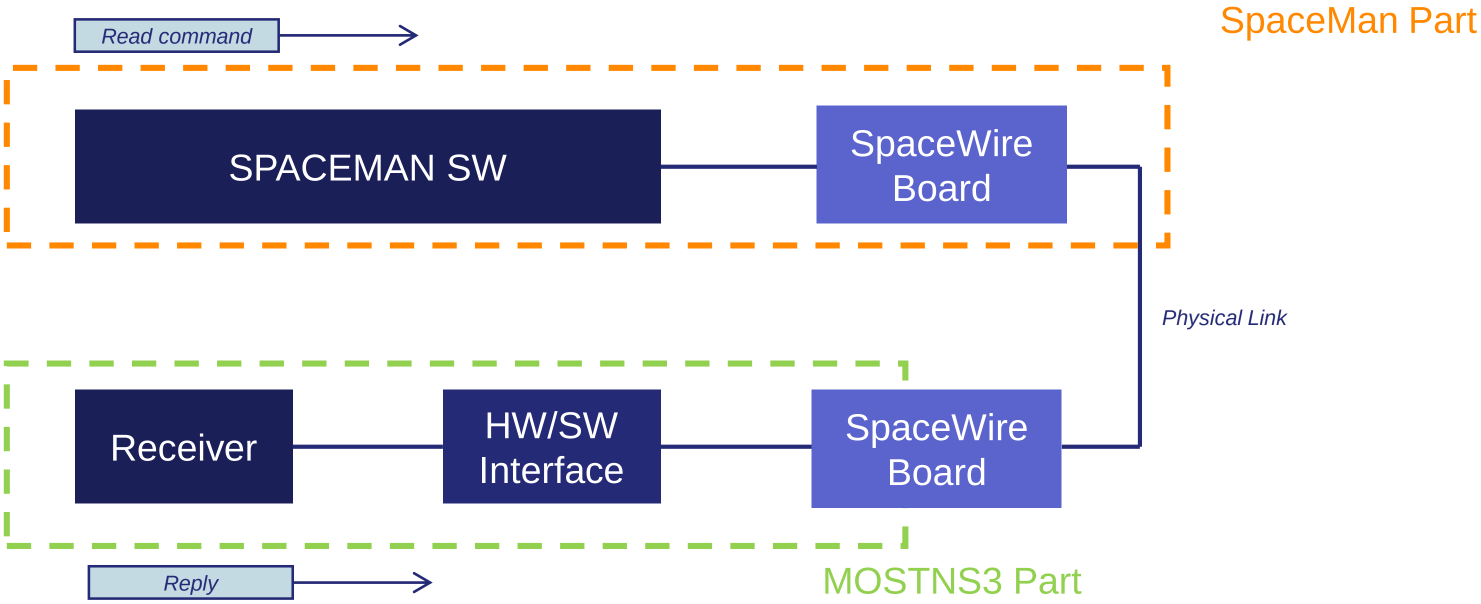


Time From Trigger	Time Delta	End A	End A Delta
-598.73126 ms		Header: FE	
-598.73121 ms	50 ns	Cargo Size: 8 bytes	50 ns
-598.73084 ms	370 ns	EEP	370 ns
-405.81462 ms	192.91622 ms	Header: FE	192.91622 ms
-405.81457 ms	50 ns	Cargo Size: 5 bytes	50 ns
-405.81435 ms	220 ns	EOP	220 ns
-203.1386 ms	202.67575 ms	Header: FE	202.67575 ms
-203.13855 ms	50 ns	Cargo Size: 7 bytes	50 ns
-203.13823 ms	320 ns	EEP	320 ns
-970 ns	203.13726 ms	Header: FE	203.13726 ms
-920 ns	50 ns	Cargo Size: 19 bytes	50 ns
0 ns	920 ns	EOP	920 ns

Basic Validation with ATOM for compatibility:

- Simple Sending
- Simple Emitting
- Loop back tests

VALIDATION WITH SPACEMAN

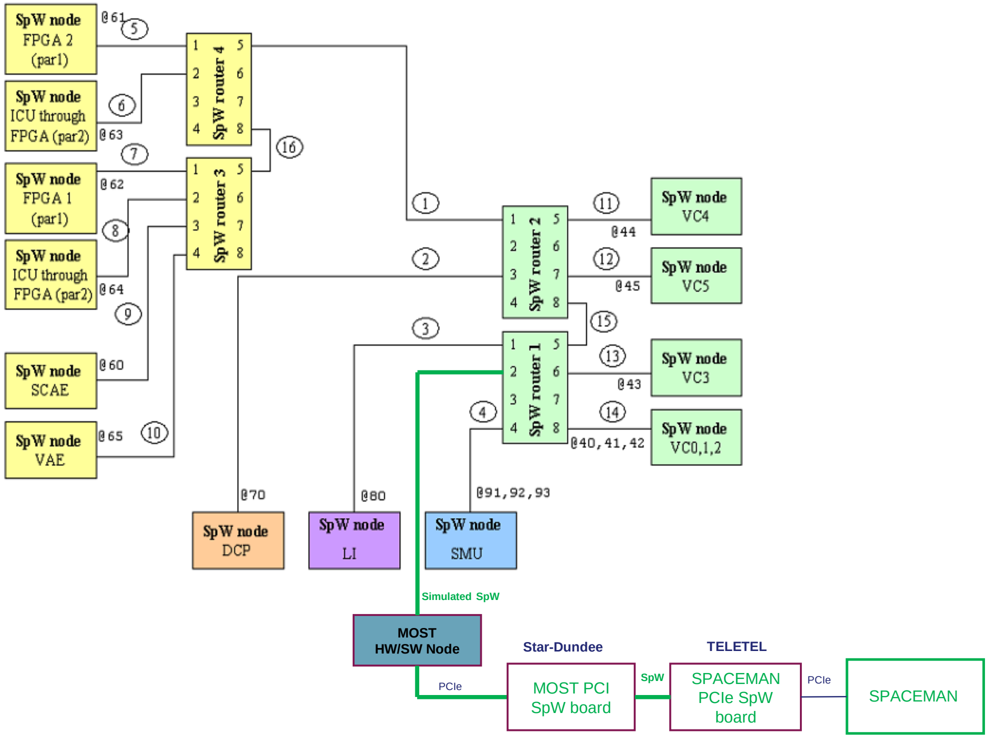


SPACEMAN and MOST are working independently but sharing information through RMAP commands

EXAMPLE OF NETWORK DISCOVERY

Simulation Logic

- SPACEMAN SW
- Connected to SPACEMAN PCIe board
- Connected in SpW to MOST PCIe board
- Linked to HW/SW Node (MOST node)
- Linked to:
  - either one Standard Node
  - Or one RMAP Node
  - Or one Router supporting RMAP



QUESTIONS?