

Remote Memory Access in Embedded Networked Systems

**Valentin Olenev,
I Korobkov,
Liudmila Koblyakova,
Felix Shutenko**

SUAI

Introduction

- ✦ A Modern Embedded Networked Systems frequently consist of a large quantity of devices (nodes and routers) allocated on different distances from each other.
- ✦ It is necessary to configure the devices, to set:
 - ✦ the routing table,
 - ✦ a device operating mode and other parameters.
- ✦ It is also necessary for devices to communicate during the operating process:
 - ✦ to update the routing table at connection or disconnection of new devices,
 - ✦ or write some information to memory of remote device (a useful data transferring from one user to another).
- ✦ In most cases it could be done by using write and read commands in appropriate program-accessible components of the remote device (e.g. memory, registers).

Thereby remote access to memory is one of the most important tasks in Embedded Networked Systems.

SpaceWire. RMAP

- ✦ There are many approaches to solve the task of remote memory access.
- ✦ For example, ESA (European Space Agency) in collaboration with international space agencies including NASA, JAXA and RosCosmos has designed RMAP (Remote Memory Access Protocol) protocol to support a wide range of SpaceWire standard applications.
- ✦ Let's consider in more detail what is SpaceWire and RMAP

SpaceWire

- ✦ SpaceWire technology has grown organically from the needs of on board processing applications. This Standard provides a formal basis for the exploitation of SpaceWire in a wide range of future on board processing systems.
- ✦ This Standard addresses the handling of payload data and control information on board a spacecraft.
- ✦ It is a standard for a high speed data link, which is intended to meet the needs of future, high capability, remote sensing instruments and other space missions.
- ✦ SpaceWire provides a unified high speed Data handling infrastructure for connecting together sensors, processing elements, mass memory units, downlink telemetry subsystems and EGSE equipment.

SpaceWire

- ✦ This Standard specifies the physical interconnection media and data communication protocols to enable the reliable sending of data at high speed (between 2 Mb/s and 400 Mb/s) from one unit to another.
- ✦ SpaceWire links are full duplex, point to point, serial data communication links.
- ✦ The scope of this Standard is the physical connectors and cables, electrical properties, and logical protocols that comprise the SpaceWire data link.
- ✦ SpaceWire provides a means of sending packets of information from a source node to a specified destination node.
- ✦ SpaceWire does not specify the contents of the packets of information.

RMAP

- ✦ The remote memory access protocol (RMAP) provides means for a SpaceWire node to write to and read from memory inside another SpaceWire node.
- ✦ The aim of the RMAP protocol is to standardize the way in which SpaceWire units are configured and to provide a low-level mechanism for the transfer of data between two SpaceWire nodes.
- ✦ All read and write operations defined in the RMAP protocol are posted operations i.e. the source does not wait for an acknowledgement or reply to be received.
- ✦ This means that many reads and writes can be outstanding at any time.
- ✦ It also means that there is no timeout mechanism implemented in RMAP for missing acknowledgements or replies.
- ✦ If an acknowledgement or reply timeout mechanism is required it must be implemented in the source user application.

RMAP

Advantages of RMAP protocol:

- ✦ simplicity of implementation:
 - ✦ 4 write commands: (acknowledged or non-acknowledged, verified or non-verified),
 - ✦ 1 read command
 - ✦ 1 read-modify command that allows to fulfill atomic operations;
- ✦ wide application and flexibility:
 - ✦ can be used for a wide range of tasks, interpretation of address fields of a RMAP-command depends on the developer
- ✦ reliability
 - ✦ available CRC-check of header and body of the command
- ✦ transmission bulk of data
 - ✦ can transfer to 16 Mbytes of the data in one command;
 - ✦ supports 32 and 40-bit address space.

RMAP Software Implementation

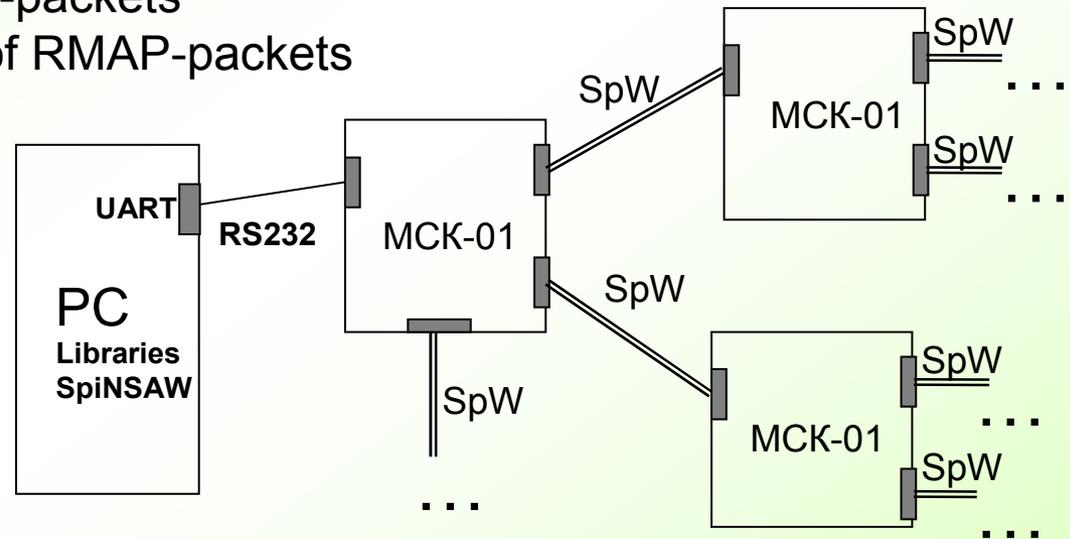
- ✦ RMAP is used for configuration of any remote router or node in a SpaceWire network

- ✦ RMAP software implementation consists of 2 parts:

 - ✦ *Initiator* (Generator) of RMAP-packets

 - ✦ *Target* (Handler, processor) of RMAP-packets

- ✦ Each MCK-01 router has embedded RISC core and specialized embedded software (firmware), which supports RMAP-packets processing.



- ✦ One of the MCK-01 router is connected to PC through UART, where special GUI application (like SpaceWire Administrator workstation) or Libraries work.

 - ✦ PC is the RMAP-packet generator,

 - ✦ MCK-01 firmware is RMAP-packet processor.

RMAP Software Implementation. Example

Path to Router
Source path

Routing table

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Prior	HDel
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
48	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Adaptive group routing registers

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<input type="checkbox"/> ADG_1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_3	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
<input type="checkbox"/> ADG_10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
<input type="checkbox"/> ADG_11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> ADG_12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
<input type="checkbox"/> ADG_13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
<input type="checkbox"/> ADG_14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
<input type="checkbox"/> ADG_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
<input type="checkbox"/> ADG_16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Routing table control

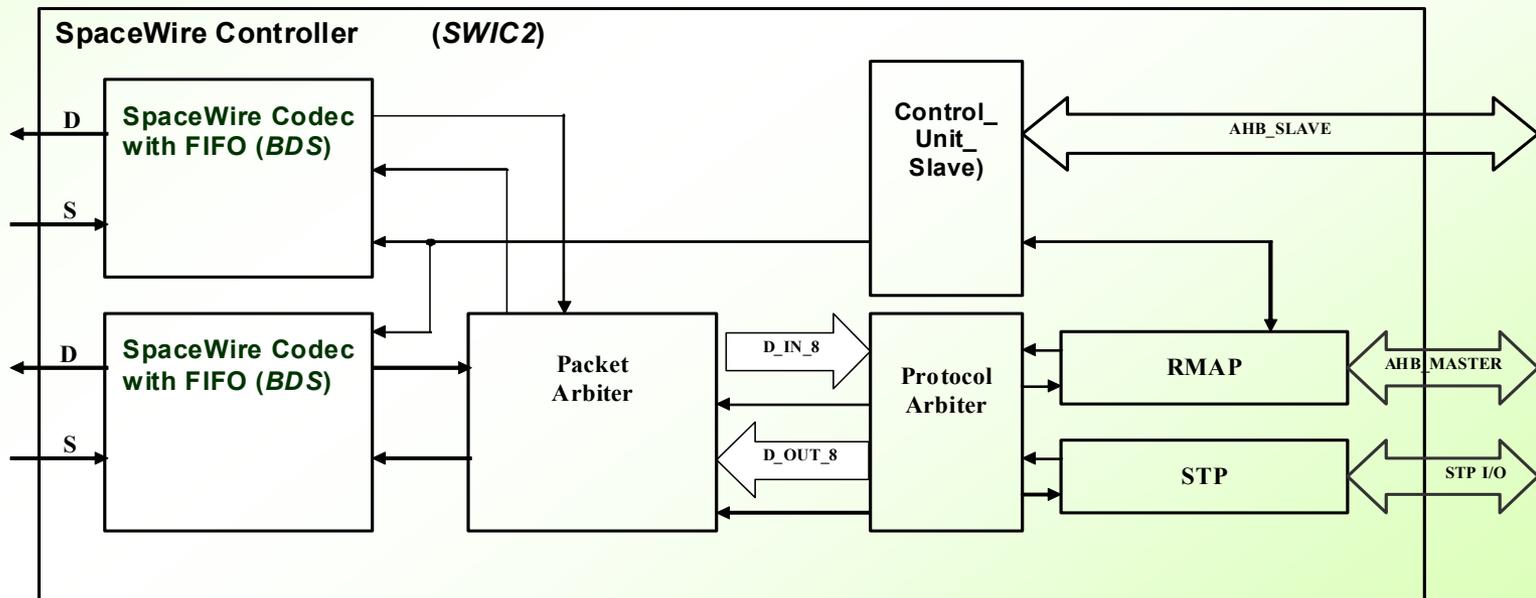
Enter strings number

Adaptive group routing registers control

SpaceWire Slave Controller

RMAP hardware implementation

- Include 2 SpW Codecs
- Internal Control Unit (AHB slave, optional)
- RMAP component compatible with ECSS-E-50-11 Draft F (do not support Read-Modify-Write command)
- STP component
- IRQ interface



SpaceWire Slave Controller

RMAP hardware implementation

- supports read and write commands only;
- send write-reply and read-reply packets only;
- the HW module can be implemented as FPGA IP-Core (Spartan and Virtex family)

Characteristics:

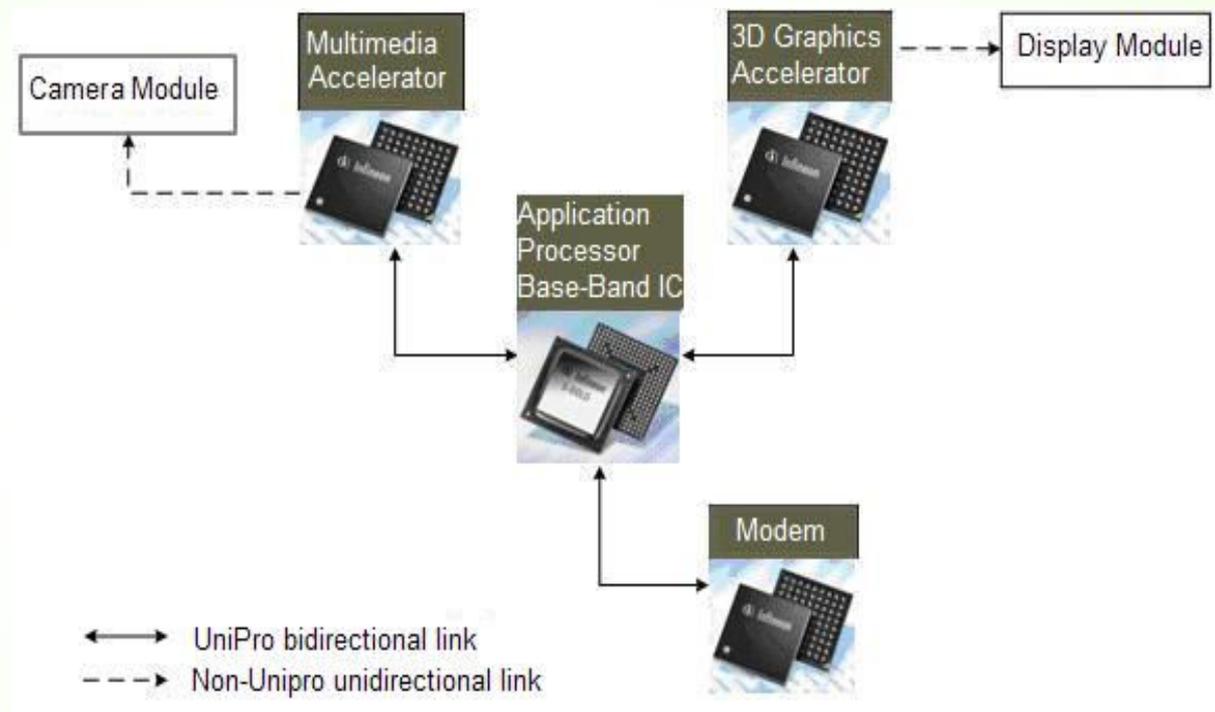
- 1300 slices (2400 LUTs) = 26% of 3s500evq100-4 Spartan-3 device
- 65 Mhz local clock of 3s500evq100-4 Spartan-3 device

SpaceWire Interface	SpW Slv	Info
iDOUT iSOUT iSIN iDIN		
System		
iRST iCLK_LOCAL iCLK_5MHz iCLK_SEND		
STP input		STP output
iSTP		oSTP
RMAP AHB master input		RMAPAHB master output
imAHB		omAHB
IRQ Interface		
iIRQ[n:0] oIRQ[n:0]		

What is UniPro

✦ UniPro is a multilayer unified protocol on the high-speed serial interface for integration different devices and components with portable systems, such as application processors, modems, video cameras, displays and other peripheral units, and associated with them different traffic types

✦ Developer of UniPro is the MIPI Alliance – an open standards development organization with over than 150 member companies, mainly from the mobile communications industry.

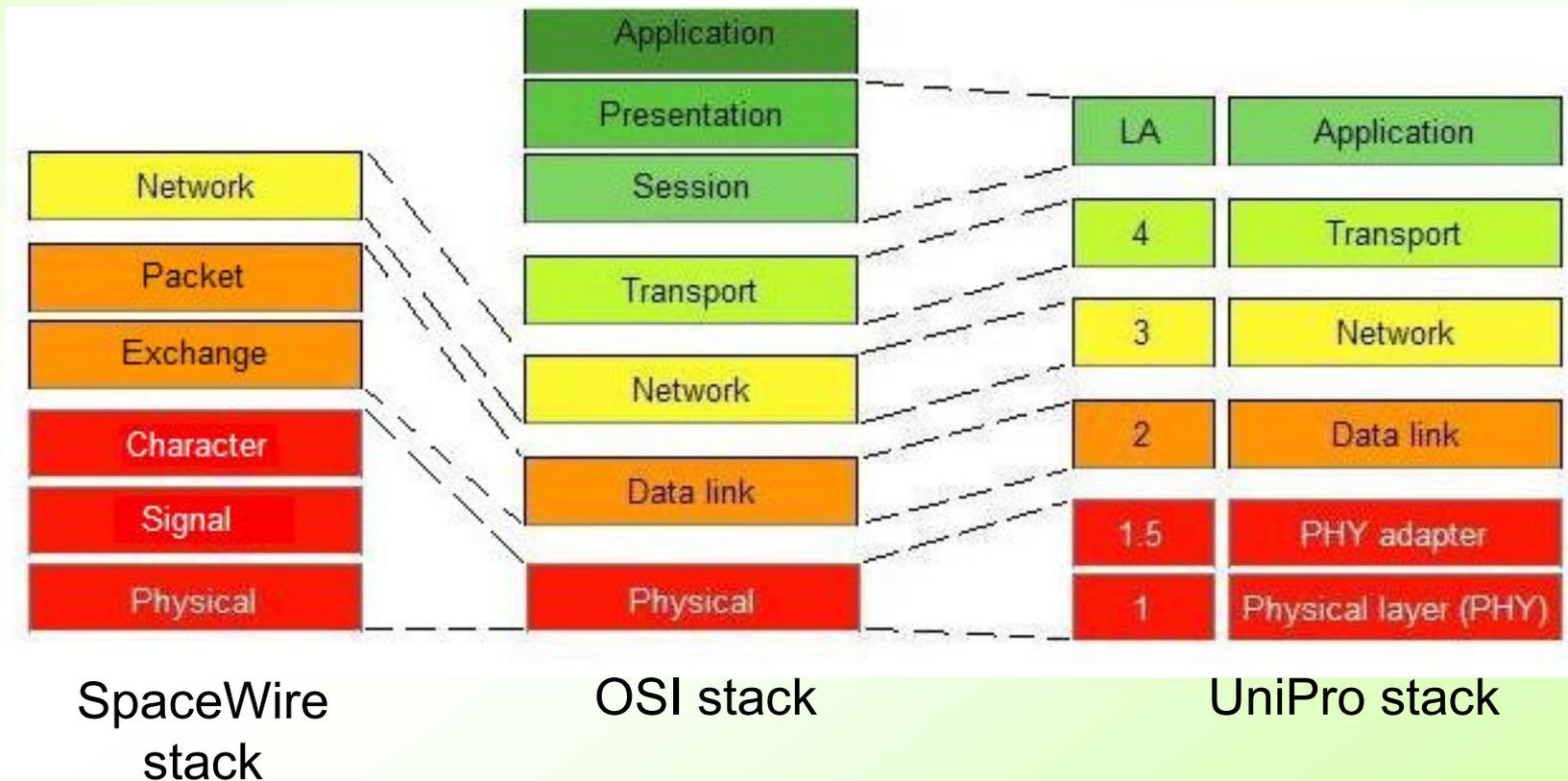


UniPro Key Features

- ✦ **High data rate (gigabit/s)** - serial technology with a number of bandwidth scaling options
- ✦ **generic** - can be used for a wide range of applications and data traffic types
- ✦ **low-power** - optimized for small battery-powered systems
- ✦ **reliability** - data errors detected and correctable via retransmission
- ✦ **bandwidth utilization** - provides features to manage congestion and control arbitration
- ✦ **hardware friendly** - can be implemented entirely in hardware where needed

UniPro and SpaceWire Architecture

- ✦ The UniPro protocol stack follows the classical OSI reference architecture.
- ✦ The UniPro specification covers **only** Layers 1.5, 2, 3, and 4.
- ✦ The Application Layer (LA) is out of scope, because different uses of UniPro will require different LA protocols. The Physical Layer (L1) has the same situation. OSI Layers 5 (Session) and 6 (Presentation) are counted as part of the Application Layer.
- ✦ UniPro's strict layering enables it to be used for a wide range of applications

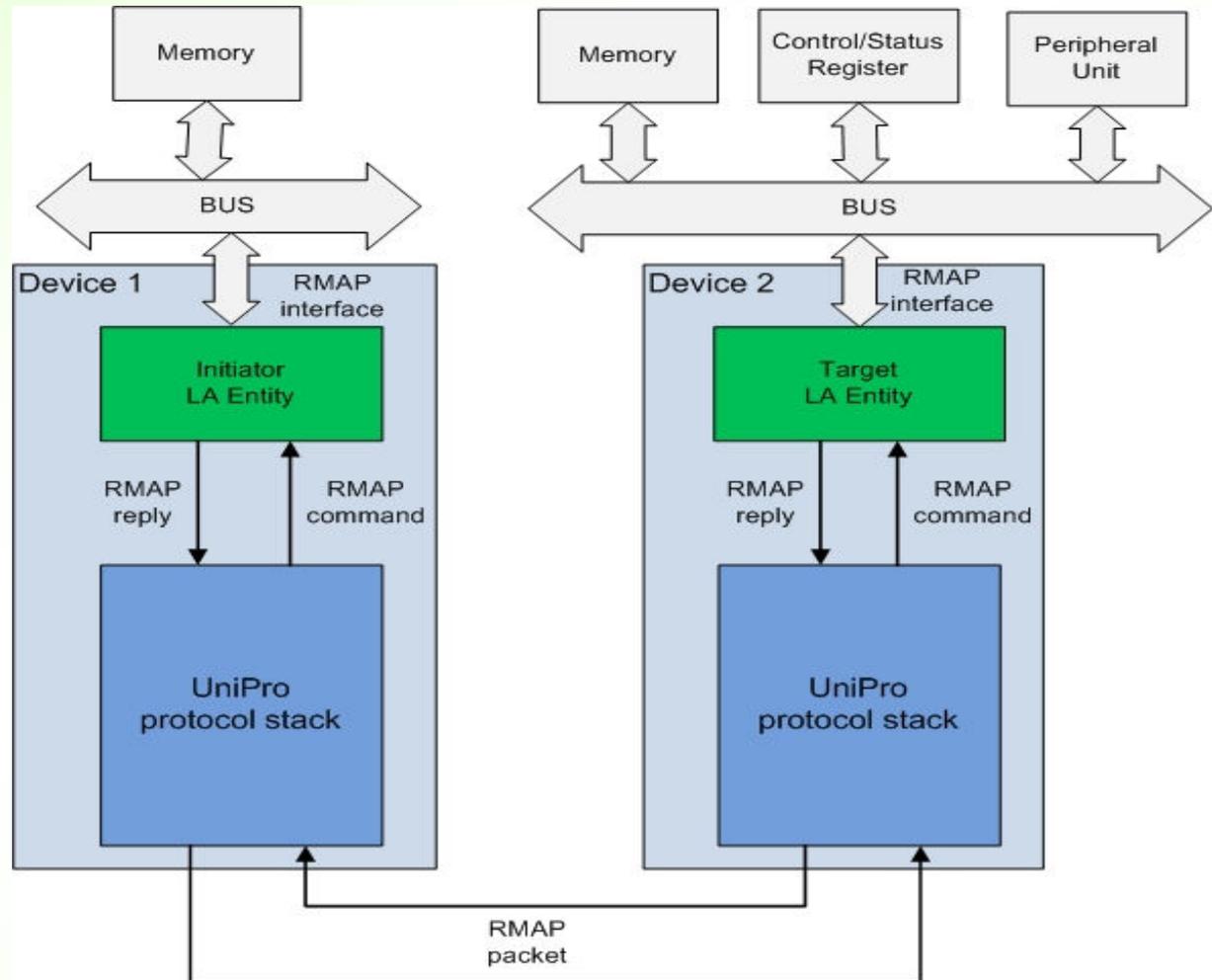


RMAP over UniPro

- ✦ So long as PIE protocol is not specified so far, it is possible to use RMAP.
- ✦ RMAP protocol will be in role of a method of remote access to memory, and LA layer will be responsible for logical processing of incoming and outgoing RMAP-commands and RMAP-replies.
- ✦ LA object will be LA Entity. It can appear both the initiator and the target of RMAP-packages.

RMAP over UniPro

- ✦ RMAP packets are created in the “Initiator LA Entity” and encoded as RMAP packets
- ✦ RMAP-packets go to Device 2 over the UniPro
- ✦ They are decoded by the “Target LA Entity” into Device 2 and data or information is passed to Memory, Control/Status Register or other Peripheral Unit after authorization of the RMAP-command.
- ✦ The “Target LA Entity” forms an RMAP reply packet which is sent to Device 1 over UniPro link. And the same actions have repeated again



Conclusion

Thereby, using RMAP protocol it is possible to implement remote access to memory on high-speed channel UniPro

Thank you