"A PROGRAMMABLE AND HIGHLY PIPELINED PPP ARCHITECTURE FOR GIGABIT IP OVER SDH/SONET"

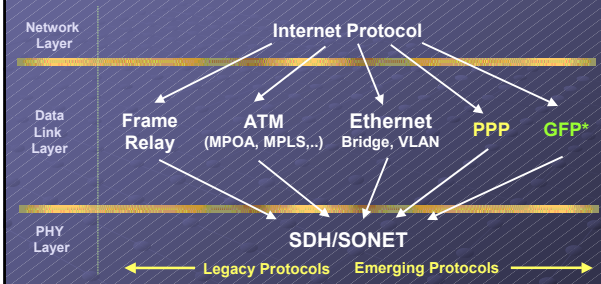
Ciaran Toal, Sakir Sezer

10th Reconfigurable Architecture Workshop 2003

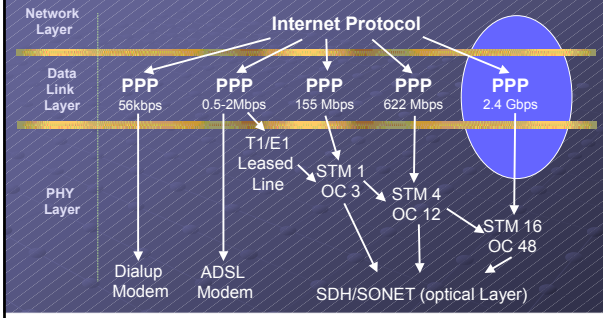
Outline

- Introduction to IP over SONET/SDH
- The Point-to-Point Protocol (PPP)
- PPP System Architecture
- PPP Data-Path
  - 32-Bit Programmable CRC Unit
  - Escape Detect and Escape Generate Units
- Synthesis and Circuit Analysis
  - Comparison between 8-bit and 32-bit implementations
- Conclusions

IP over SDH/SONET Layer 2 Technologies



Point-to-Point Protocol based Networking



The Point-to-Point Protocol

- The most efficient layer 2 protocol for encapsulating IP datagrams
- Key applications include ADSL, dialup modems, encapsulated Ethernet, Virtual Local Area Networks (VLAN), Virtual Privet Networks (VPN) etc.
- Key functions:
  - Framing and Error Control method for encapsulating data over physical layer point-to-point links.
  - Link Control Protocol (LCP) functions to establishes, negotiate configure and terminate PPP links between two network nodes.
  - Network Control Protocol (NCP) function for upper network protocols, optional for ATM, IP, Ethernet etc.

The PPP Frame Format

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

- **The PPP frame is made up of the following fields:**
  - *Flag* - A single byte which indicates the beginning or end of a frame. The flag field always consists of the binary sequence 01111110
  - *Address* - A single byte that contains the binary sequence 11111111. This is the standard broadcast address
  - *Control* - A single byte that contains the binary sequence 00000011 which requests user data transmission in an unsequenced frame
  - *Protocol* - Two bytes that identify the protocol encapsulated in the information field of the frame
  - *Payload* - Zero or more bytes that contain the datagram for the protocol specified in the protocol field
  - *Frame Check Sequence (FCS)* - Normally 16 bits (2 bytes) but can be 32 bits (4 bytes) for improved error detection

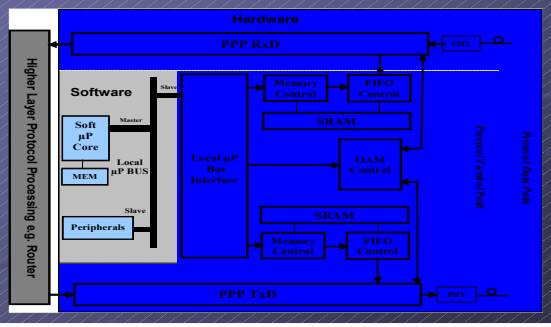
Programmable SoPC System Architecture for PPP processing

The Programmable SoPC Architecture for PPP processing is composed of three architectural independent units.

This includes :

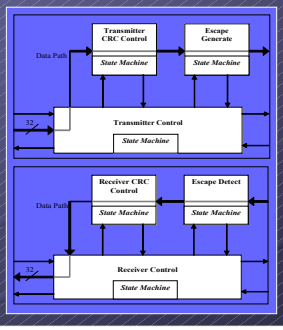
- **Protocol Data-Path Unit**
  A highly pipelined and parallel frame processing circuit
- **Protocol Control-Path Unit**
  Data path control, Register and control protocol FIFO circuit
- **Embedded Microcontroller Unit**
  Responsible for Control and management protocol processing and Data-Path configuration.

PPP SoPC System Architecture

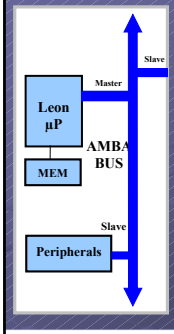


P5 Protocol Data-Path Units

- 32-bit wide circuit. Complete implementation in hardware
- operating at 78.125 Mhz offering a data transfer rate of 2.5 Gbps
- TxD and RxD units are independent, parallel and pipelined
- Focus of presentation is on this unit. 8-bit version has also been implemented



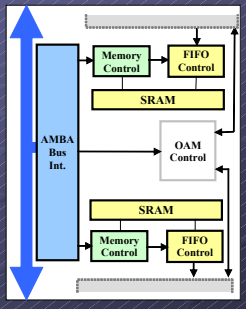
Embedded Microprocessor Unit



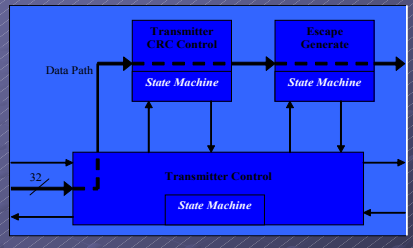
- Based on a soft IP core processor (e.g. Nios Microblaze or Leon).
- Processes the majority of the control and management protocols and control / housekeeping functions.
- Interfaced to a local bus via a standard embedded bus architecture e.g. AMBA
- All implemented functions are software based and can be reprogrammed without interrupting the data transmission path.

Protocol Control-Path Unit

- Handles the interaction between the embedded processor and the Layer 2 protocol data path
- Composed of an Operation Administration and Maintenance (OAM) unit,
- Accommodates a transmitter and a receiver FIFO for temporary storage of control protocols.

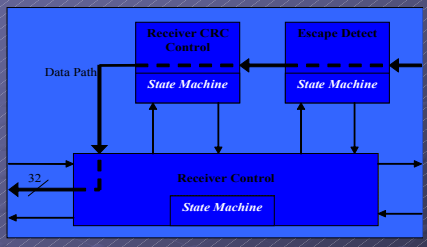


The P5 TxD Data-Path



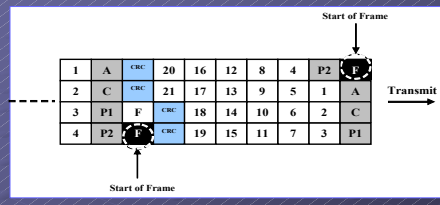
- Consists of 3 main units, a control unit, a CRC processing unit and an Escape generator unit.
- The data-path is heavily pipelined.

The P5 RxD Data-Path



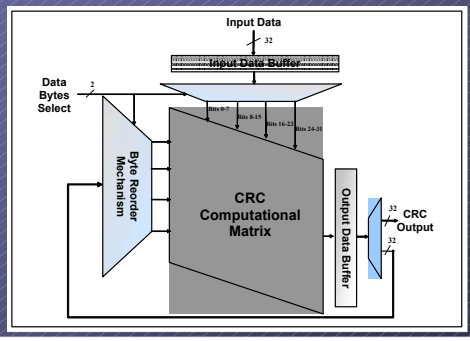
- Essentially the reverse operation of the TxD.
- The circuit of each block is very different from the TxD sub-blocks.

Configurable 32-Bit parallel CRC computation



- CRC unit must be able to process 1, 2, 3 or 4 input bytes
- The need for configurability requires further pipelining
- Wait state insert for backpressure must be included

Configurable 32-Bit CRC Calculator Circuit



PPP Frame Delimiter

- Require safe-guard for any instances of flag character **0x7E** outside of the flag fields
- Escape character **0x7D**
- Flag character 0x7E replaced by escape character 0x7D and original data with sixth bit complicated.

e.g. consider the following data sequence:

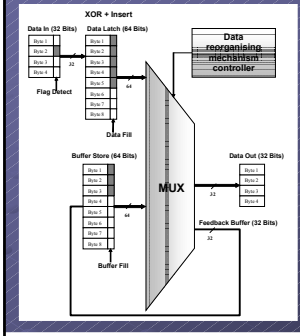
32-Bit Escape Generate

- Consider 4 bytes of data to be processed
- Byte location 2 contains a flag character 0x7E

Byte Location 1 (7-0)	A1	Byte Location 1 (7-0)	A1
Byte Location 2 (15-8)	7E	Byte Location 2 (15-8)	7D
Byte Location 3 (23-16)	12	Byte Location 3 (23-16)	5E
Byte Location 4 (31-24)	8C	Byte Location 4 (31-24)	12
		Extra Byte	8C

- The Escape generator inserts a 0x7D (Esc) character.

32-Bit Escape Generate Circuit



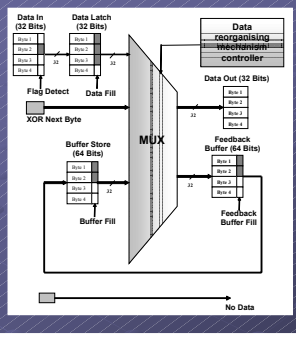
- Extra pipelining is added to reduce critical path.
- Data reorder mechanism is introduces to insert Escape characters and to rearrange the 4 byte data-path frame data.
- Buffer is introduces to enable full cycle back-pressure mechanism after inserting 4 Escape characters.

32-Bit Escape Detect

- Consider 4 bytes at the receiver
- Byte location 2 contains byte escape character 0x7D

Byte Location 1 (7-0)	A1	Byte Location 1 (7-0)	A1
Byte Location 2 (15-8)	7D	Byte Location 2 (15-8)	7E
Byte Location 3 (23-16)	5E	Byte Location 3 (23-16)	8C
Byte Location 4 (31-24)	8C	Byte Location 4 (31-24)	Empty

32-Bit Escape Detect Unit



- Extra pipelining is added to reduce critical path.
- Data reorder mechanism is introduces to eliminate inserted Escape characters.
- Buffer is introduces to enable full cycle wait states.

8-Bit System Synthesis Results

8-Bit System						
	Pre-layout Synthesis			Post-layout Synthesis		
XCV50 -4	95.3 MHz	184 LUTs	84 Register	79.5 MHz	130 Slices	191 LUTs
XC2V40 -6	128.4 MHz	179 LUTs	84 Register	91.5 MHz	124 Slices	185 LUTs

- 8-bit PPP meets required speed of 78.125 Mhz with Virtex and Virtex II technology
- Virtex II enables considerable speed-up over Virtex.
- Critical paths analysis revealed the same number of LUTs for both technologies. Thus speed-up is achieved because of the technological advantage of Virtex II

32-Bit System Synthesis Results

32-Bit System						
	Pre-layout Synthesis			Post-layout Synthesis		
XCV600 -4	73 MHz	2641 LUTs	841 Register	65 MHz	1208 Slices	2563 LUTs
XC2V1000 -6	125.9 MHz	2230 LUTs	689 Register	78.66 MHz	1144 Slices	2157 LUTs

- The 32-bit PPP implementation meets the required speed of 78.125 Mhz with Virtex II technology only.
- Again, critical paths analysis revealed the same number of LUTs for both technologies.
- The 32-bit implementation is ×11 times larger than the 8-bit implementation.

Escape Generate Unit Synthesis

Escape Generate Module				
	32-Bit Implementation		8-Bit Implementation	
XCV40 -6	492 LUTs (96%)	168 Register (32%)	22 LUTs (4%)	6 Register (~1%)

Size increase x 22.36

- 32-bit Escape Generate unit requires 22 times more combinational logic and 28 times as many flip-flops as the 8-bit version.
- Therefore the large size increase for the 32-bit PPP is due to decisional logic and data reordering mechanisms included within the CRC and Escape units .

Conclusions

- We have shown that a programmable Layer 2 network processing for 2.5 Gbps throughput rate, including control protocol processing is feasible using the latest SoPC technology.
- The circuit study has revealed that the 32-bit PPP implementation is 11 times larger than 8-bit version.
- Further analysis has shown that this increase is mainly due to the byte sorter and buffering mechanisms included in the 32-bit design which are heavy in combinational logic.
- Programmability of PPP processing is achieved by reconfiguring the programmable logic blocks and by reprogramming the firmware of the embedded processor.

Future Work

- ASIC Implementation of the current PPP SoPC architecture as a complete SoC (System on a Chip) solution.
- Investigating new technologies and systems architectures for programmable / configurable network processing.
- Investigating trade-offs using off-the-shelf embedded processor and FPGA technology for network processing.
- Development of a new generation of programmable packet processing elements by combining configurable logic with custom processing technology.