Institute of Experimental Physics - Slovak Academy of Sciences
Kosice, Slovakia

CERN - European Organization for Nuclear Research
Geneve, Switzerland

The SPD router design
(Draft)

M. Krivda

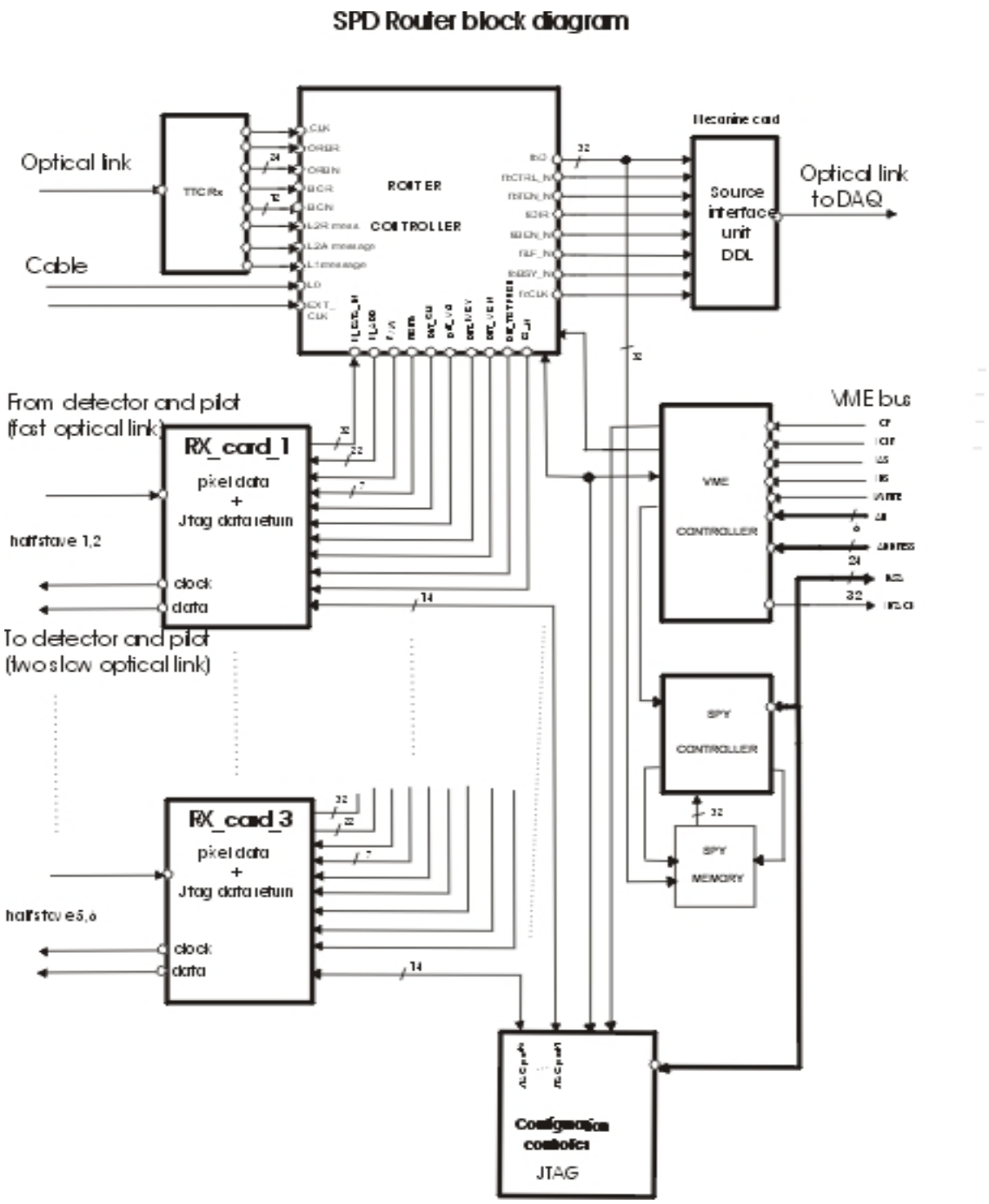


Fig.1 The proposal of SPD router electronics.

This block diagram (Fig. 1.) shows the proposal of router electronics as it is created by electronics group from Institute of Experimental physics in Kosice in collaboration with CERN pixel group.
Source interface unit (**SIU-DDL** [1]) and Link mezzanine card (**LMC** [2]) components will be delivered from CERN group and interfaced to router board as mezzanine cards.

Main task of 9U VME Router module is provided the interface between the on detector electronics and the data acquisition system (DAQ). Each Router module will have three Link mezzanine card (LMC) in order to connect all the links required for the operation of half sectors. Each LMC will read two half-staves on the same side of SPD. One channel (LMC) will have three optical fibre links, one to receive data being returned from the Pixel Detector and Pilot Chip MCM and a two for the transmission of trigger control signals, JTAG parameter and configuration data to the Pilot Chip MCM. The main data flow arrives from the Pixel detector sectors. Data are formatted, filtered and labelled with trigger identification information. So the router creates and arranges queues from different physical events. The output of the router will be coupled to the DDL as well as a dual port memory, where it can be acessed via VME. Into dual port memory flow copies of the data transmitted to the DDL. These data will be accessible via the VME port of the Router to allow online monitoring of the data sended to DAQ. There must be software synchronization for retrieving this data. The VME is assigned to DCS and detector configuration tasks. If there is a process reading this extra memory, it must be synchronized with the rest of the software in order to avoid access conflicts.

The Router sends commands to the Pilot Chip, the Pilot chip MCM and Link receiver card. The commands are trigger control signals, RESET signals and signals for communication with Link receiver mezzanine board. On the other side the router receive pixel data (the structure of pixel data is defined in document “Row data format of one SPD sector”, Working document: Design of Nov 16, 2000), Jtag return data and control signals (status register) from Link receiver card.

The router merges the hit data from 3 Link mezzanine card (6 half stave) into one data block and stores them into a memory where the data wait to be transferred to the ALICE data acquisition (DAQ) over the detector data link DDL. Each data block will be prefixed by the header, which will be added to the corresponding pixel data hit. The header is defined in document “Data Format over the ALICE DDL “, R. Divià, P. Vande Vyvre, ALICE-INT-2002-10 V 1.7).

Basic functionality of Router

In idle state router is waiting for L0. It is used for synchronization and checking of right protocol. Than router is waiting for L1 (L1 timeout is 6 μs). If router doesn’t receive L1 in a right time, it asserts error. The error can only be cleared by a reset signal. After the receipt of L1 router is waiting for L2Y or L2N.

If router receives L2Y then will check an event ready bit, link ready bit and status register for errors in all 3 LMC in sequence from LMC 0 to LMC 2. If there is assertion of this bits, a presence of at least one event is ready and there were no errors present, then the data will be read out from corresponding multi-event buffer (on LMC) after the start and the end addresses have been read. Using these addresses the event data will read out and router will assert the flush event register in order to delete corresponding event data from link receiver memory.

If router receives L2N, than it will assert the flush event register directly and data will not be read out from dual-port memory on LMC.

Before merging the data for the DDL output must be done the check on the data synchronism (event numbers from all the 6 half staves).

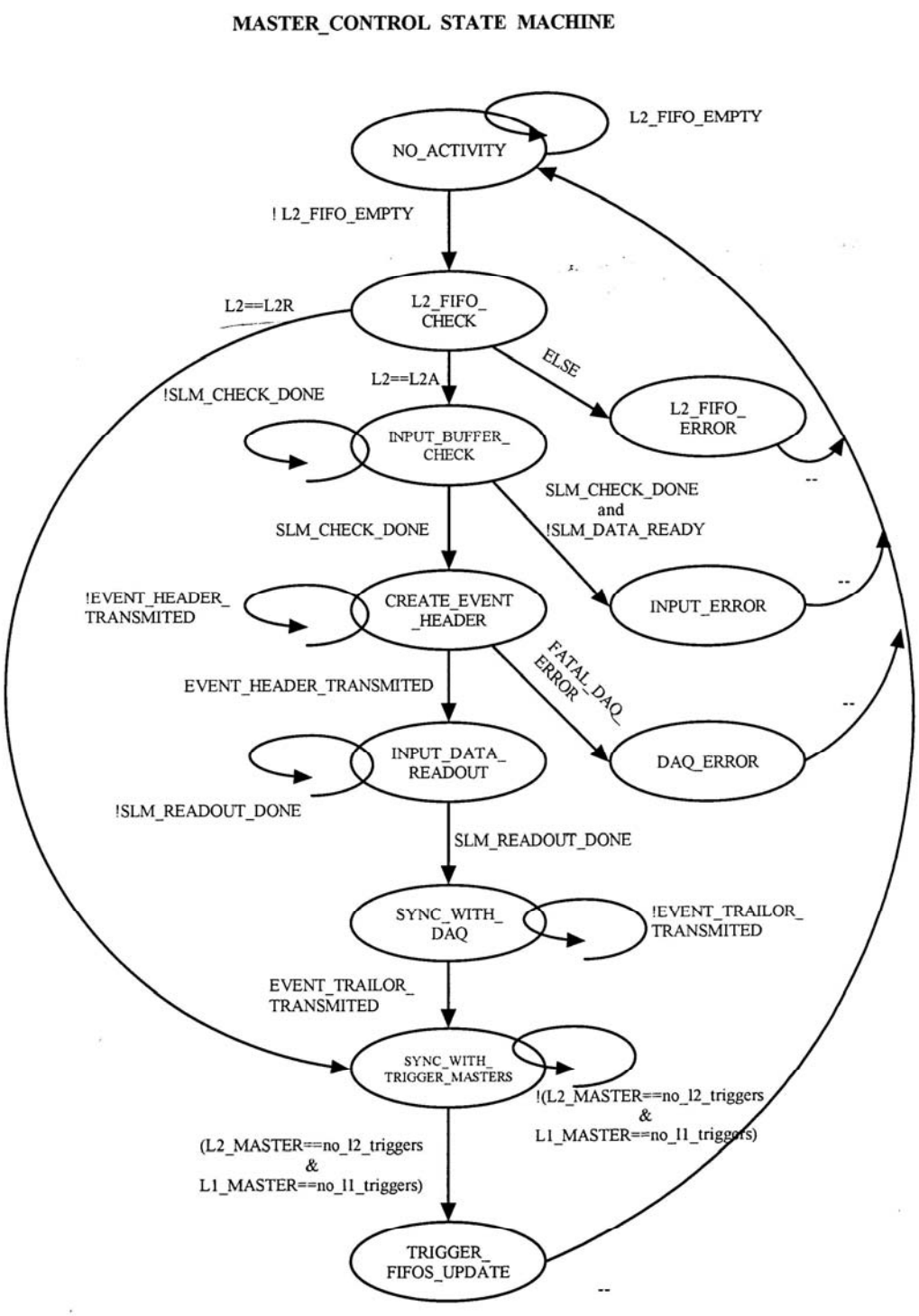


Fig. 2 Master control state machine

Overall multiplexing data process follows “MASTER CONTROL STATE MACHINE” (Fig.2).

Configuration controller

On the router board will be placed **Configuration controller of SPD – (JTAG)** which is being used to load configuration of Pixel chips and monitoring (voltage, temperature, ….) on detector. It will observe 8 JTAG ports, 6 ports for 6 half staves, one for monitoring temperature, votlage and one for spare. JTAG data are sent to detector over slow optical link and receive data being returned from Pixel Detector over fast optical link.

In IDLE state JTAG controller is waiting for Execution_start command**,** which will start JTAG_master_state_machine. The input buffer FIFO_IN must be filled written data, before one asserts Execution start. The failure of doing so will result in ERROR_1 (bit 9 in STATUS register). JTAG controller processes macroinstructions, which are stored in FIFO_IN, and data are sent to corresponding JTAG channel. Returned data (from TDO) are stored in FIFO_OUT and can be read out via VME port.
ERROR_4 (bit 31 in STATUS register) appears when FIFO_OUT is full and one want to write data to this buffer.
ERROR_3 (bit 17 in STATUS register) appears when controller wants read trailer and there is not trailer.
ERROR_2 (bit 11 in STATUS register) appears when controller wants read control_word_ 2 and there is header.

SPY controller

SPY controller maintains a copy of data samples sent to DAQ system. These data are stored in the SPY memory and can be accessed via VME interface. The stored data can be used for monitoring and test purposes. The SPY memory itself is high-speed (3.3V 32Kx36) synchronous pipelined dual-port RAM.

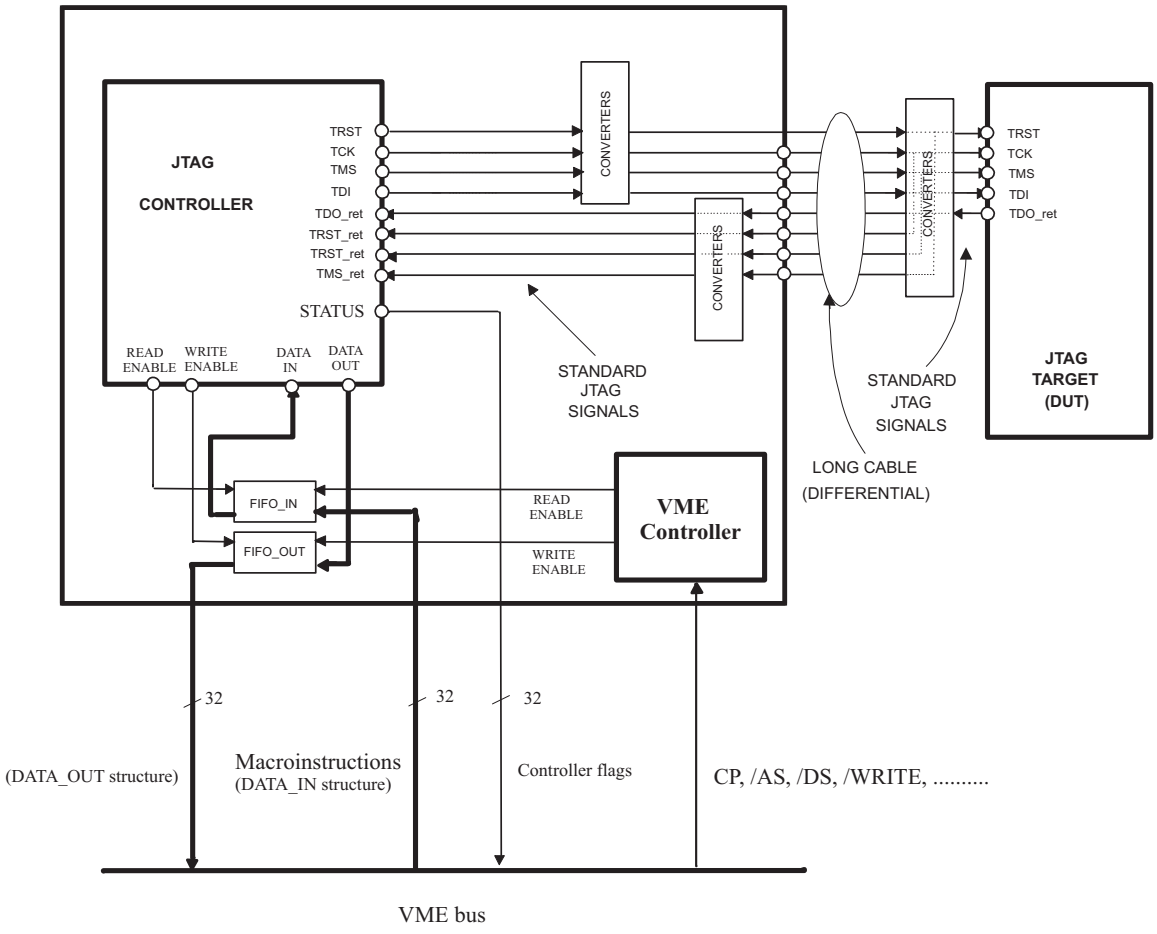
Embeded macros

Marcos add an extra functionality to the JTAG controller. We can write all commands to the controller via the VME, but in many cases we just repeat the same well defined commands, therefore we will store such macros in flash memory within the router and send them to the controller just by a simple VME write to a control register. A main interest it is mostly for a configuration data. Here is an example why this would be usefull: After powering on the Alice1 chips, the power consumtion goes significantly up and we need to write a correct configuration to the chip. Once we power on the whole SPD, this might be a problem (in some cases this might event induce an interlock). The software can start configuring the SPD ladder per ladder, but... It can get stuck due to many reasons, which will leave SPD in an unpleasant state. If we could store some "typical" settings in a flash memory, the software could just send along with the power=on command also a simple command to router which will in turn load the chips with the configuration stored in the flash memory. This is not necessarry the most correct configuration, the detector will be anyway reconfigured by the software before the data taking. The idea is just to load the chips with something which will keep the power

consumption within acceptable limits untill the software will be available for the definitive and correct configuration.

Router will execute macros for monitoring MCM, temperature and voltage automatically in regular intervals too. In this way it get the MCM voltage information in a very convenient way. A received information will store into output memory, where it will still refresh with new informations.

JTAG controller



JTAG controller block diagram

Fig. 7. Functional block diagram of JTAG controller.

This block diagram shows the fundamentals of Configuration controller designed by electronics group IEP SAS Kosice. The controller will be incorporated into SPD Router 9U board.

TRST signal:

- This is a hardware signal, which is not implemented in all hardware components. An optional soft reset (TMS = 1 for at least 5 TCK cycles) can be used to reset by this devices.

TCK signal:

- Clock frequency can be set in control word 1.

TMS signal:

- This signal can be optionally delayed using the on-board delay units. Implementation of this feature reflects the requirement of ALICE 1 chip.

TDI signal:

- Similar to TMS, this signal can be internally delayed.

Signals TRST_ret, TCK_ret, TMS_ret are received independent state machine in Jtag_controller and can be connected with Jtag_controller board using jumper, but only for short cable. Otherwise the **TRST_ret, TCK_ret, TMS_ret** signals must be interconnected on chip with TRST, TCK, TMS! It assures that all delays will be the same as TDO_ret.

JTAG_controller memory

- Data (macroinstruction) are stored in FIFO_IN memory and output data read out from FIFO_OUT memory through VME bus.
- FIFO_IN and FIFO_OUT are dual port FIFO memories. Read and write uses different clocks – read and write cycles are independent.

JTAG_ports

- Data can be sent to one of the 7 ports implemented in JTAG controller.
- Port can be selected in Control word 1 (bits 10..8) and number of port is 0 – 6. Default port is port number 0.
- Every macroinstruction can be sent to different port. Each macroinstruction can be also scanned at different speed.

JTAG_instructions

- There are 3 fundamental instructions executed by the controller:
  1. IR Scan
  2. DR Scan
  3. RESET
- Type of instruction can be set in control word 1 (bits 7.. 0) as
  - 2 for RESET instruction
  - 4 for DR Scan
  - 5 for IR Scan

Clock Speed

- Main controller operates at a frequency of 10 MHz.
- Clock for JTAG channels can be set in control word 1(bits 15..12), up to 5 MHz.
- Default clock speed is the slowest - 312,5 kHz.
- Option for clock speed if clock number is 4 - 5 MHz
  3 - 2,5 MHz
  2 - 1,25 MHz
  1 - 625 kHz
  other - 312.5 kHz.

Data format

- Macroinstructions and output data are 32 bits words.
- The macroinstructions are defined by DATA_IN_STRUCTURE and the output data are defined by DATA_OUT_ STRUCTURE. It is same structure, but data are different.
- Structure of the header and trailer is identical for both the macroinstructions and the output data.

Header and trailer:

Bits	**hex format**
31..0	FFFFFFFF

Control word 1:

Bits	**Description**
31..16	Not used
15..12	Clock speed for JTAG channel
10..8	Number of port
7..0	Type of instruction

Control word 2:

Bits	**Description**
31..28	Number of bits in last word (0..31)
27..0	Number of word in macroinstruction (1..26,8 M)

End of block macroinstructions:

Bits	**hex format**
31..0	FFFFFFFF
31..0	FFFFFFFF

Execution of header followed immediately by a trailer will send the JTAG controller to IDLE state and will wait for another Execution_start.

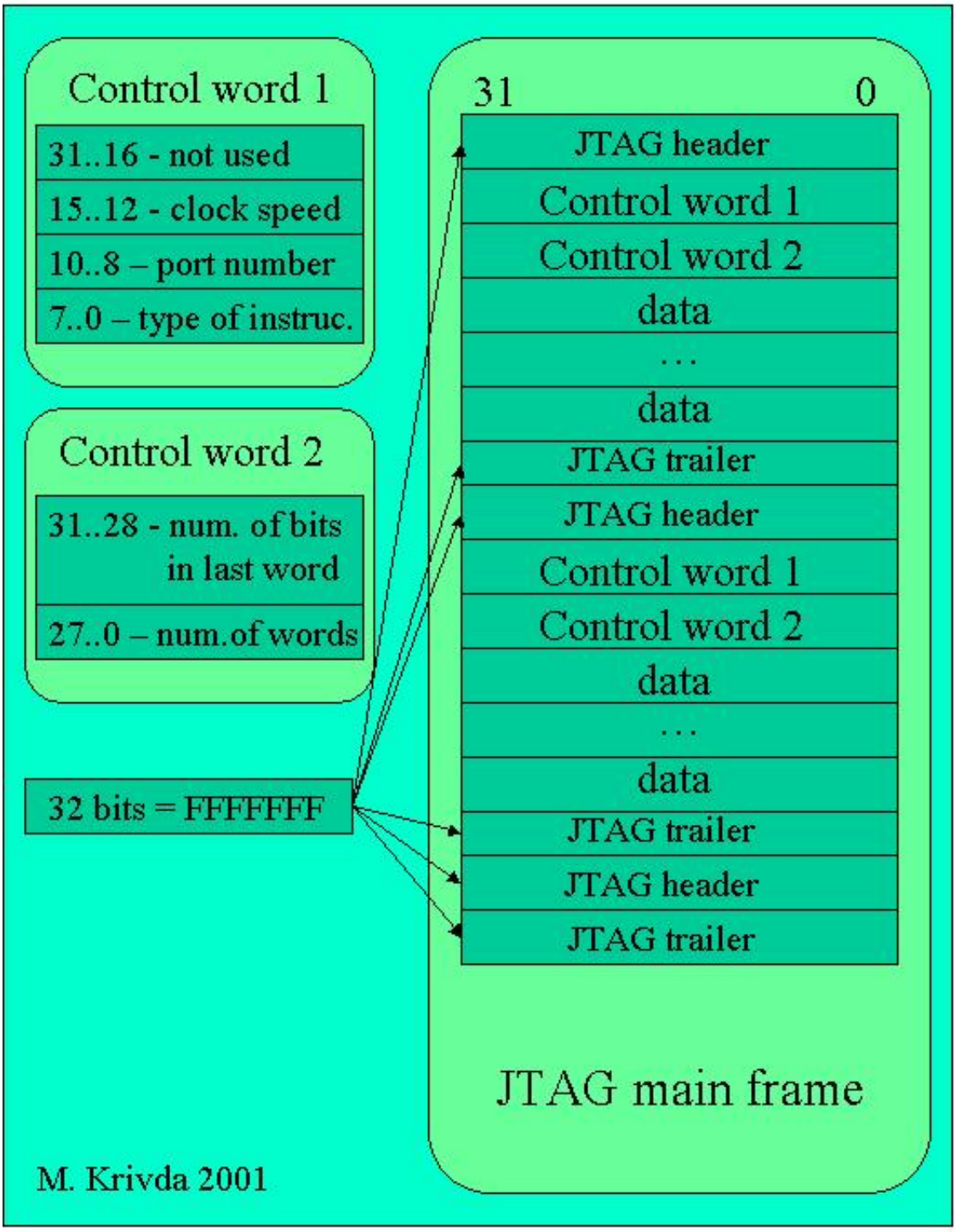


Fig. 8. Data in and out structure.

STATUS register

- STATUS register is a 32 bits word describing the state of the controller and errors from the last operation.

Meaning of individual bits is:

JTAG status word	Bit	Indicated status
IDLE	0	Controller is IDLE
I_EXEC_STARTED	1*	Start of master state machine
I_EXEC_STARTED_FIFO	2*	Init FIFO
READ_INSTR_HEADER	3*	Reading header
HEADER_CHECK	4*	Header check
END_OF_MEMORY_CHECK	5*	Indicates empty FIFO_IN on reading
READ_INSTR_CODE	6*	Reading control word 1
END_OF_INSTRUCTION_QUEUE	7*	END_OF_INSTRUCTION_QUEUE
RESET_I_ADDRESS_COUNTER	8*	Indicates reset after ERROR 1-3
ERROR_1	9	In FIFO_IN are not data
READ_WC_AND_LAST_BIT_COUNT	10*	Reading control word 2
ERROR_2	11	Wrong data after reading READ_WC_AND_LAST_BIT_COUNT
READ_I_DATA	12*	Reading data
DATA_PROCESSING	13	Sending data to JTAG channel
END_OF_INSTRUCTION_CHECK	14*	End of instruction check
TDO_END_TEST	15*	TDO end testing
READ_I_TRAILER	16*	Reading trailer
ERROR_3	17	Wrong data after reading READ_I_TRAILER
END_OF_JTAG_SEQUENCE_CHECK	18*	Checking end of JTAG sequence
EOM	19*	FIFO_OUT is full
ShiftDR_TDO_IN	20*	ShiftDR_TDO_IN
ShiftIR_TDO_IN	21*	ShiftIR_TDO_IN
Seq_Run_Test_Idle	22	Indicates Run Test Idle state
Seq_SelectDRScan	23*	Sequencer_SelectDRScan
Seq_SelectIRScan	24*	Sequencer_SelectIRScan
Capture	25*	Sequencer Capture
Shift	26*	Sequencer Shift
Exit_1	27*	Sequencer Exit_1
Pause	28*	Sequencer Pause
Exit 2	29*	Sequencer Exit 2
Update	30*	Sequencer Update
ERROR_4	31	Controller can’t write data to FIFO_OUT, because FIFO_OUT is full.

* These bits are used for debugging of JTAG controller and they will change to useful information.

Design of JTAG_controller is compliant with IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std 1149.1-1990).

The VME JTAG assumes A24/D32 transfers. All data from described registers are 32 bits, but only few bits are really necessary from some registers.

VME address of board can be set on board by hexadecimal switch:

IC 22 (4 valid bits)
IC23 (only 1 valid bit - most significant in the address: A23 bit)

VME address allocation:

Write data	Write-only	0x074000
Read data	Read-only	0x070000
Execution_start	Write-only	0x078000
Read status JTAG controller	Read-only	0x068000
RESET_JTAG_controller	Write-only	0x07c000
RESET_FIFOs	Write-only	0x08c000
Reset_JTAG_channel_0	Write-only	0x064000
Reset_JTAG_channel_1	Write-only	0x060000
Reset_JTAG_channel_2	Write-only	0x05c000
Reset_JTAG_channel_3	Write-only	0x058000
Reset_JTAG_channel_4	Write-only	0x054000
Reset_JTAG_channel_5	Write-only	0x052000
Reset_JTAG_channel_6	Write-only	0x050000

REFERENCES

[1] M. Burns et al., **The ALICE Silicon Pixel Detector Readout System**,
6 th Workshop on Electronics for LHC Experiments
[2] A. Kluge, **The Read-Out system of the ALICE pixel detector**
[3] JTAG specification
[4] Wade D.Peterson, The VMEbus Handbook, third edition, 1996
[5] ALICE/98-21 ALICE DETECTOR DATA LINK,
Hardware Guide for Front-end Designers.
[6] TTC web site: http://www.cern.ch/TTC/intro.html
[7] P.Jovanovic, Local Trigger Unit – Preliminary Design Review, 1.9.2002
[8] ALICE CTP web site: http://www.ep.ph.bham.ac.uk/user/pedja/alice/