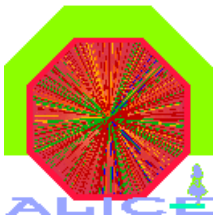

TTC Interface Test Board (TTCit)

PRELIMINARY SPECIFICATION

Revision 2.2 26 July 2004

DOCUMENT HISTORY

Version	Date	Editor	Comments
Draft 0.1	24.08.2000	P. Jovanović	First draft
Draft 2.0	17.12.2003	A. Jusko	
Draft 2.1	22.07.2004	P. Jovanović	Updated version; emphasis on the technical details; examples from the LTU design included.
Draft 2.2	26.07.2004	P. Jovanović	Section **2.5** (*TTC data FIFO*) expanded; *latched error* added (section **2.4.6**); description of the **L0** input included (section **2.9**).

TABLE OF CONTENTS

1.0 INTRODUCTION 1
1.1 Preface 1
1.2 Document overview 2
1.3 Definitions and acronyms 2
1.4 Signal name abbreviations 2
1.5 References 3
2.0 TECHNICAL SPECIFICATION 5
2.1 VME interface 5
2.2 TTCit PCB 5
2.3 Front panel 5
2.4 LED signals 7
2.4.1 VME indicators 7
2.4.2 Clock indicators 7
2.4.3 Signal indicators 8
2.4.4 Error indicators 8
2.4.5 Other LEDs 10
2.4.6 General requirements 10
2.5 TTC data FIFO 11
2.6 SnapShot memory 12
2.7 Scalers 13
2.8 ScopeProbe signals 13
2.9 L0 synchronization 14
2.10 FPGA programming and configuration 14
2.11 DC/DC converters and voltage regulators 14
2.12 Power supply - fuses and filtering 15
2.13 FPGA decoupling 15
2.14 TTCrx or TTCrm? 15
2.15 TTCrx connection to the FPGA 19
2.16 Spare connections and test points. 19
2.17 FPGA connections 19
2.18 TTCit board identification 20

1.0 INTRODUCTION

1.1 Preface

For some time, the design and the development of the TTC Interface Test board (TTCit) has been hampered by a lack of an up-to-date and sufficiently detailed technical specification. Apart from a very early draft [7], made obsolete by numerous changes of the CTP requirements, and a relatively recent, but too general revision [8], the ALICE collaboration, and the Birmingham group in particular, have not so far produced a document that could serve as a base and a starting point for the TTCit board project.

The following text is an attempt to "bridge the gap" and "kick-start" the development work by the Košice group. It is written in a hurry, with few explanations and the emphasis on the technical details; it is intended for a person doing the design work rather than a general reader; it is certainly a *personal view.*

Many references have been made to the ALICE Local Trigger Unit (LTU). It is partly because the board is the most recent Birmingham project, but, mainly, because there is a *significant similarity* between the TTCit and the LTU design concepts. The TTCit designers should make a good use of the circuit design, and the firmware and software development done for the LTU.

The TTCit designers need to get familiar with the ALICE CTP [1] and the LTU board [2] in particular - the systems that generate signals and data that are transmitted over the TTC optical link; the designers should *know by heart* the operation and the technical specification of the TTCrx chip [3] that receives and decodes the information. The designers could also benefit from the LTU board schematics [6].

1.2 Document overview

The present section (***Introduction***) gives a brief account of the purpose and scope of the document, explains its structure, defines terms and acronyms and lists cited documents. The following section (***Technical Specification***) lists the TTCit requirements and includes a number of details, options, suggestions, examples, *etc*. that, it is believed, could help the TTCit board designers.

1.3 Definitions and acronyms

BC Bunch Crossing (clock) - the 40.08 MHz clock, locked to the LHC machine cycle, used to synchronise the pipeline processing system.

CTP Central Trigger Processor [1] - electronic system that receives inputs from ALICE trigger sub-detectors and generates, in each bunch crossing, **L0**, **L1** and **L2** yes/no trigger decisions for all sub-detectors.

DAQ (ALICE) Data Acquisition system.

FIFO First-in-first-out memory (buffer)

L0 Level-0 trigger (signal).

L1 Level-1 trigger (signal).

LTU Local Trigger Unit (board).

LVDS *Low Voltage Differential Signalling* - a standard differential signal format.

PLL Phase locked loop.

RoI Region of Interest (option, logic, data) [14].

TTC Timing, Trigger and Control (system) [9].

TTCit TTC Interface Test (board).

TTCrm TTC Mezzanine Evaluation Board [10].

TTCrx Timing, Trigger and Control Receiver ASIC [3].

1.4 Signal name abbreviations

BCID[12..1] ***BC** identifier* word, part of the event identifier.

ESR *Enable Segmented Readout* flag, part of the RoI option.

1.5 References

[1] *ALICE Central Trigger Processor - Preliminary Design Review,* current version available on the ALICE CTP web site [4].

[2] *ALICE Local Trigger Unit - Preliminary Design Review,* current version available on the ALICE CTP web site [4].

[3] J. Christiansen et al., *TTCrx Reference Manual*, current version available on the TTC web site [5].

[4] ALICE CTP web site: (ALICE→Projects→Trigger), or, directly, http://www.ep.ph.bham.ac.uk/user/pedja/alice/.

[5] TTC web site: http://www.cern.ch/TTC/intro.html.

[6] Circuit diagrams of the LTU board; contact *P.Jovanovic@bham.ac.uk*.

[7] *TTCit Preliminary Specification*, Draft 0.1, 19 July 2000; contact *P.Jovanovic@bham.ac.uk*.

[8] *TTCit Preliminary Specification*, Draft 2.1, 17 December 2003; contact *A.Jusko@bham.ac.uk*.

[9] B.G. Taylor, *TTC Distribution for LHC Detectors*, IEEE Trans. Nuclear Science, Vol. 45, June 1998, pp. 821-828.

B.G. Taylor, *LHC Machine Timing Distribution for the Experiments*, Proceedings of the Sixth Workshop on Electronics for LHC Experiments, Crakow, Poland, 11-15 September 2000, pp.312-317.

[10] *TTCrm Mezzanine Evaluation Board*, data available on the TTC web site [5].

[11] *LTU SnapShot Memory*, contact *P.Jovanovic@bham.ac.uk*.

[12] *LTU Flash Memory Interface*, contact *P.Jovanovic@bham.ac.uk*.

[13] *LTU FPGA Configuration*, contact *P.Jovanovic@bham.ac.uk*.

[14] *Region of Interest Interface*, proposal approved by the ALICE TB on 14 May 2002; available on the ALICE CTP web site [4].

2.0 TECHNICAL SPECIFICATION

2.1 VME interface

VME64 slave (J1 and J2 connectors with 96 pins):

A24 (performs 24-bit address transfers across the backplane);
D32 (performs 32-bit data transfers across the backplane);
Responds to the following Address Modifier codes (AM[5..0]):

- H"39" - nonprivileged data access,
- H"3A" - nonprivileged program access,
- H"3D" - supervisory data access,
- H"3E" - supervisory program access.

VME base address:

- A[23..16] - H"8A";
- A[15..12] - Set by the 4-bit **Board Address** dial;
- A[11..2] - 10 bits used to decode up to 1k of *local addresses*;
- A[1..0] - 0.

2.2 TTCit PCB

Form factor: 6U.

Number of layers: 8.

1. signal layer (component side),
2. ground plane,
3. signal layer,
4. main power supply plane (+3.3V),
5. auxiliary supply plane (+5V, +1.5V, *etc*.),
6. signal layer,
7. ground plane,
8. signal layer (solder side).

Note: Six layers might be sufficient, but 8 enable a "cleaner" layout at a negligible cost increase.

2.3 Front panel

The TTCit front panel is shown in **Figure 2.3**.

The ScopeProbe connectors (A,B) are LEMO 50Ω (size 00, SCEM 09.46.11.180.6). The L0 connector is LEMO 2-pin, SCEM 09.31.28.070.5.

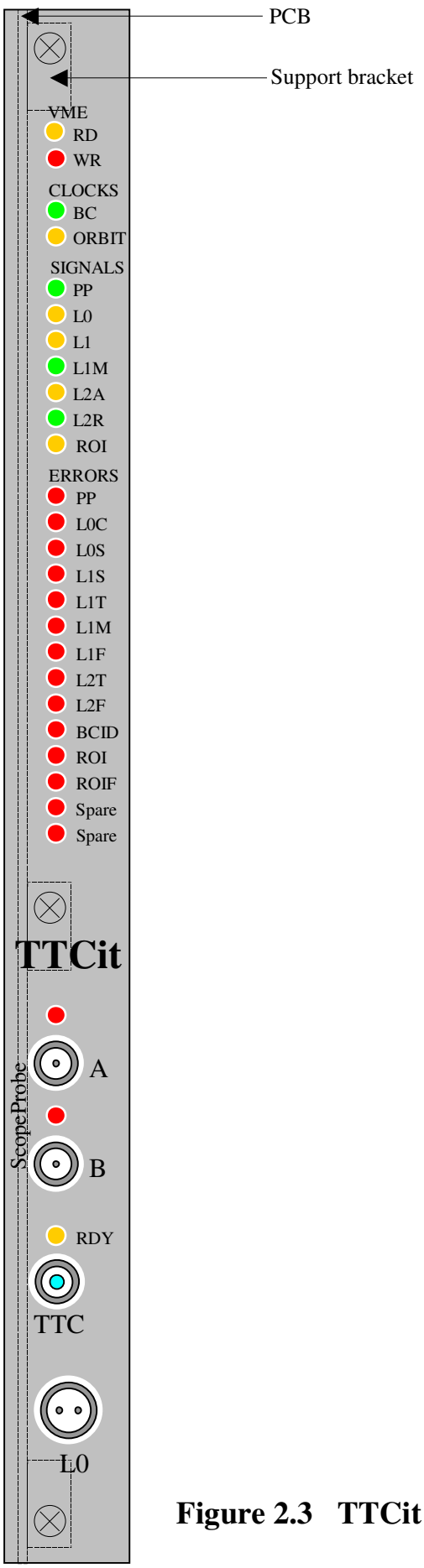

Figure 2.3 TTCit front panel - *proposal*

The LEDs are 3mm/T1 HLMP-1700 (red), HLMP-1719 (yellow) and HLMP-1790 (green), driven from the FPGA *via* a 470Ω resistor (sufficiently bright, no dazzle).

2.4 LED signals

2.4.1 *VME indicators*

RD - VME read

Any VME *read* in the selected board address range (A[23..0], AM[5..0]).

WR - VME write

Any VME *write* in the selected board address range (A[23..0], AM[5..0]).

2.4.2 *Clock indicators*

BC - BC clock status

Complement of the **BC Error** signal. In normal conditions, the LED is permanently on.

The **BC Error** signal indicates:

- **BC** clock not present;
- **BC** frequency outside the 38MHz to 42MHz range;
- a missing **BC** pulse;
- a glitch.

Note: A logic circuit that generates the **BC Error** signal has been developed for the LTU board.

ORBIT - ORBIT signal status

Complement of the **ORBIT Error** signal. In normal conditions, the LED is permanently on.

The **ORBIT Error** signal indicates that the ORBIT signal is not present, or its frequency is too low.

Note: A logic circuit that generates the **ORBIT Error** signal has been developed for the LTU board.

2.4.3 *Signal indicators*

PP - Pre-pulse signal

L0 - L0 signal

L1 - L1 signal

L1M - L1 Message

Transmission of the *L1 Message* header.

L2A - L2a Message

Transmission of the *L2a Message* header.

L2R - L2r Word

Transmission of the *L2r Word*.

ROI - RoI Message

Transmission of the *RoI Message* header.

2.4.4 *Error indicators*

PP - Pre-pulse error

Transmission of the **Pre-pulse** signal in a bunch-crossing different from the programmable *Pre-pulse BC* interval/range.

L0C - Calibration L0 error

Transmission of the calibration **L0** signal in a bunch-crossing different from the programmable *Calibration BC* interval/range.

L0S - Surplus L0

Transmission of any surplus **L0** signal during the interval between a correctly received **L0** and its corresponding *L1 decision time*; the *L1 decision time* is programmable.

L1S - Surplus L1

Transmission of any **L1** signal not preceded by a corresponding **L0** signal. The *L1 decision range* is programmable.

L1T - L0-L1 time violation

Transmission of any **L1** signal outside of the *L1 decision* BC interval. The *L1 decision interval* follows the corresponding **L0** transmission with a programmable delay.

Note: By programming, the **L1S** and the **L1T** errors can be made identical. On the other hand, in a general case, they could be set to identify different error patterns - a missing **L0**, or an **L0/L1** transmission jitter.

L1M -L1 Message error

No *L1 Message* header transmitted following an **L1** signal; the time-out interval is programmable. Also, an *L1 Message* header transmitted without a corresponding **L1** signal.

Note: An *L1 Message* header delayed beyond the time-out shall generate 2 error signals - at the end of the time-out interval; and when it is transmitted. Separate LEDs/signals could be used, if necessary.

L1F -*L1 Message* format error

Incomplete or extended *L1 Message* - too few or too many *L1 Message* data words; or a data word not preceded by the header word.

Note: The error-checking state machine is triggered by either a programmable time-out following the transmission of the header word, or by a transmission of another header word.

L2T - L1-L2 time violation

No *L2a Message* header/*L2r Word* transmitted following an **L1** signal; the time-out interval is programmable. Also, an *L2a Message* header/*L2r Word* transmitted without a corresponding **L1** signal.

Note: An *L2a Message* header/*L2r Word* delayed beyond the time-out shall generate 2 error signals - at the end of the time-out interval; and when it is transmitted. Separate LEDs/signals could be used, if necessary.

L2F -*L2a Message* format error

Incomplete or extended *L2a Message* - too few or too many *L2 Message* data words; or a data word not preceded by the header word.

Note: The error-checking state machine is triggered by either a programmable time-out following the transmission of the header word, or by a transmission of another header word.

BCID -BC Identifier error

The *BC Identifier* in the *L2a Message/L2r Word* different from the content of the TTCrx *BC Counter* at the transmission of the corresponding **L1** signal.

ROI -RoI Message error

No *RoI Message* header transmitted following an *L1 Message* with the **ESR** flag asserted; the time-out interval is programmable. Also, a *RoI Message* header transmitted without a corresponding *L1 Message* with the **ESR** flag asserted.

Note: A *RoI Message* header delayed beyond the time-out shall generate 2 error signals - at the end of the time-out interval; and when it is transmitted. Separate LEDs/signals could be used, if necessary.

ROIF -*RoI Message* format error

Incomplete or extended *RoI Message* - too few or too many *RoI Message* data words; or a data word not preceded by the header word.

Note: The error-checking state machine is triggered by either a programmable time-out following the transmission of the header word, or by a transmission of another header word.

2.4.5 *Other LEDs*

Spare LEDs

Provided for future use. At present, could be used for *FIFO full* status; SnapShot memory *Sampling Mode*; software driven LED; *etc.* .

A,B - *ScopeProbe* LEDs

Display the status of the selected *ScopeProbe* output.

RDY - TTCrx READY status

Driven by the TTCrx **TTC READY** signal.

Note: Two more LEDs should also be considered: LOCK and INIT. They indicate the locked state of the FPGA PLL and successful completion of the FPGA configuration (FPGA signal INIT_DONE), respectively. During the development and the testing of the LTU board, the indicators were very useful; the SMD LEDs have been mounted in the central area of the board, seen only from the component side.

2.4.6 *General requirements*

All the LEDs:

- are permanently *on,* if the corresponding signal is permanently asserted;
- are permanently *off,* if the signal is permanently cleared;
- *flash off* for ~0.3s, if the signal is asserted, but makes a short transition to the clear state;

- *flash on* for ~0.3s, if the signal is cleared , but makes a short transition to the asserted state;
- *toggle* on (~0.3s) and off (~0.3s), if the signal is continuously changing state.

Note: A logic circuit (FPGA firmware) that provides the described LED display has been developed for the LTU board.

It shall be possible to *individually disable* (turn and keep off) any LED from the *Signals* and *Errors* groups.

When investigating infrequent/intermittent errors, it might be more convenient to use the *latched error* instead of the *real-time* error signals. The *latched error* is set at the first occurrence of the corresponding error signal and remains asserted until cleared by the control software. The *latched error/real-time error* selection shall be individually programmable for each error signal; the control software *clear* shall also be performed by individual bits. The software readout of the *latched error* signals could be a useful debugging option.

When asserted, a programmable bit, the *LED Test*, causes all the LEDs to continuously toggle. This is a quick way of testing the operation of the LEDs and their control electronics.

2.5 TTC data FIFO

FIFO capacity: 1k 20-bit words.

The FIFO interface logic (FPGA firmware) writes into the FIFO upon transmission of

- **Pre-pulse**,
- **L1** (TTC *Channel A*),
- any TTC *individually addressed command* ([3], Chapter **4**).

The FIFO is read by the control processor (*FIFO Data* read word).

The *FIFO Data* word - bit allocation:

- 16 TTC data bits,
- **Pre-pulse**.
- **L1**.
- Accumulated **L0**.
- Accumulated **Any Error**.

"Accumulated" stands for "*one or more signals received since the last write into the FIFO*"; **Any Err**or is the *OR* of all enabled error signals.

The 16 bit of data stored into the FIFO upon the transmission of the **Pre-pulse** shall read 0.

The 16 bit of data stored into the FIFO upon the transmission of the **L1** shall include the corresponding content of the TTCrx *BC Counter*.

The **L1** transmission can coincide with the transmission of the individually addressed command; the appropriate queuing/buffering is required in order to store both into the FIFO.

Note 1: The FIFO is implemented using the FPGA internal memory blocks. For the proposed capacity, 5 M4K blocks are required (out of 52 available). Both the capacity of the FIFO and the word format can be easily changed by modifying the FPGA firmware.

Note 2: The 16 TTC data bits provide for all the CTP messages/words - they are uniquely identified with their TTC address codes (see [2], section **3.6.1**). The 16 bits also enable the TTCrx *Error Dump* and *Configuration Register Dump* instructions (see [3], Chapter 4); they both should use the same "Reserved for the CTP" code (8-11), each with the unique DQ[3..0] identifier and the corresponding 8 bits of data.

The *FIFO Status* word, read by the VME processor, should include *FIFO Empty*, *FIFO Nearly Full* and *FIFO Full* status bits.

Note 3: In order to speed-up/to simplify the control software, in might be convenient to have the *FIFO Status* bits *also* added (in parallel) to the *FIFO Data* word. "Also" is emphasised because the presence of an independent *FIFO Status* word is essential.

A *test mode*, when the VME processor both writes into the FIFO and reads from it, would be desirable for debugging of the board and the FPGA firmware.

Note 4: In order to simplify the control of the SnapShot memory (see the next section), in might be convenient to add to the *FIFO Data* word the **Error Detected** signal. The signal is asserted by a programmable *OR* of all the enabled error signals and automatically cleared following the *FIFO Data* read. The **Error Detected** signal could be used by the control processor to halt the data recording by the SnapShot memory in the *Before* mode.

2.6 SnapShot memory

Memory capacity: 1M 18-bit words.

Bit allocation:

- **Orbit**,
- **Pre-pulse**,
- **L0**,
- **L1**,
- Serialized TTC data (16 serial bits),
- 12 error signals,
- spare bit.

The memory should operate in two modes:

- (sampling) *After* mode,
- (sampling) *Before* mode.

Note 1: In the *Before* mode, the sampling is stopped by the CPU command; the command could be prompted, for example, by detection of the **Any Error** signal in the FIFO readout. An additional option that ends the sampling automatically at the occurrence of the **Any Error** signal should also be considered.

A *test mode*, when the VME processor both writes into the SnapShot memory and reads from it, would be desirable for debugging of the board and the FPGA firmware.

Note 2: On the LTU board, the Cypress SSRAM chip CY7C1382B-133AC has been used for a very similar application; both the interface firmware and the control software have been developed and tested; the description of the interface operation is also available [11]. It might be convenient to re-use the available firmware and software.

2.7 Scalers

The list of signals to be scaled with *32-bit* counters:

- **Pre-pulse**,
- **L0**,
- **L1**,
- *L1 Message*,
- *L2a Message*,
- *L2r Word*,
- *RoI Message*.

The list of signals to be scaled with *10-bit* counters (*with overflow prevention*):

- **Orbit Error** (see section **2.4.2**),
- 12 error signals (see section **2.4.4**).

All the counters shall use the *Gray* code in order to enable the readout "on the fly"; during the VME readout, the content of the selected counter shall be automatically converted into the binary format.

Note: The final list of signals to be scaled shall emerge during the detailed design of he TTCit board.

2.8 ScopeProbe signals

The ScopeProbe option enables an easy oscilloscope access to a number of internal signals; the corresponding LEDs provide a quick visual check of the signal status; the option has been implemented on the LTU board. The circuit is described in [2], section **3.13**.

A list of internal signals connected to the *ScopeProbe* outputs shall emerge during the detailed design of the CTP boards.

2.9 L0 synchronization

The **L0** input is optional; the signal is required to generate the following error signals: **L0C**, **L0S**, **L1S**, **L1T** (section **2.4.4**). The **L0** input is connected to any of the three identical **L0** outputs of the corresponding LTU board ([2], section **3.5.5**). The signal format is LVDS; positive input is the pin 1 (marked with the red dot) of the 2-pin connector (SCEM 09.31.28.070.5).

The *phase adjustment* of the **L0** input signal in respect to the local (TTCrx/FPGA) **BC** clock shall be done "manually"; the phase shall be checked with an oscilloscope and, if required, the input shall be delayed 12.5ns (half of the **BC** period) by clocking it first with the negative clock edge, and then re-clocked it with the "working", positive edge. The FPGA firmware must support the described delay option.

Note: A more "clever" - automatic or semi-automatic - procedure might emerge during the detailed design. Its implementation would just require the FPGA firmware upgrade.

2.10 FPGA programming and configuration

Among a number of possible options, the following proposal offers easy upgrading of the FPGA firmware. Such flexibility is highly desirable since frequent modifications are to be expected and most of the TTCit boards will be used by the collaboration institutes at their sites.

The EEPROM based *VME controller* FPGA shall be programmed *in situe*, *via* the 10-pin *ByteBlaster* connector (JTAG), driven from a PC that runs the FPGA development software (ALTERA's *Quartus II*).

The SRAM based, *board logic* Cyclone FPGA needs to be configured upon power-on. The configuration data, the *Raw Binary File* (.rbf) generated by the FPGA development software, shall be stored, *via* the VME bus, into the on-board flash memory, and loaded into the FPGA either *automatically* - upon power-on, or *on request* from the control CPU.

Note 1: The capacity of the flash memory should provide for a simultaneous storage of at least a couple of different configurations (283kB each); no application of this feature has been envisaged so far, but it might become a desirable option in the future. The Am29LV081B-70EC flash memory with the capacity of 1M 8-bit words has been used on the LTU board.

Note 2: The proposed scheme has been implemented on the LTU board. The flash memory interface and the FPGA configuration controller are both implemented on the *VME controller* FPGA; the FPGA firmware and the control software have both been developed and tested; a detailed description of the operation of both circuits is available [12][13].

2.11 DC/DC converters and voltage regulators

A "home-made" DC/DC converter for the +3.3V supply could be used, but it needs first to be prototyped and thoroughly tested, and only then implemented

on a PCB; special attention must be paid to the noise generated by the circuit; a necessary filtering must be defined and tested.

Note 1: Alternatively, ready-made and fully specified commercial converters could be used - they are available and cheep. On the LTU board, a 3.3V/6A Texas Instruments PT6441A has been used; a new design might consider the PTH05000W circuit.

The +1.5V FPGA core supply could be generated using a voltage regulator rather than a DC/DC converter; the regulators are simpler and less noisy.

Note 2: On the LTU board, the LMS1585AIT-1.5 regulator has been used (with the appropriate heat sink).

2.12 Power supply - fuses and filtering

The main +5V power supply, taken from the VME backplane, should have a protection fuse.

Note: On the LTU board, a polymer-based self-resetting 7A **Polyswitch RGE700** (Raychem) has been used.

At the +5V input of the board, a low pass filter should be used, made with an inductance and a rather high capacitance (low ESR tantalum capacitors). An inductive filter should be used in front of the DC/DC converter in order to prevent the “kick-back” of the input current. The PLL supply of the FPGA should also be filtered with an LC circuit.

2.13 FPGA decoupling

The decoupling of the FPGAs is critical and is done best with the SMD ceramic capacitors mounted as close as practical to the corresponding pins on the solder side of the PCB. As an example, **Figures 2.13.1** and **2.13.2** show the layout of the decoupling capacitors for the two types of FPGAs on the LTU board.

Another example, **Figure 2.13.3** shows the layout on the LTU board of the +1.5V supply plane and the distribution of the FPGA PLL supply.

2.14 TTCrx or TTCrm?

The availability and the support for the TTCrm test board [10] are both unclear and uncertain. A more prudent approach would be to “copy” the board connections directly onto the TTCit PCB and avoid the “mezzanine” configuration.

Also, the TTCrm board, even if it were soldered rather than plugged-in, *would exceed a single slot hight*.

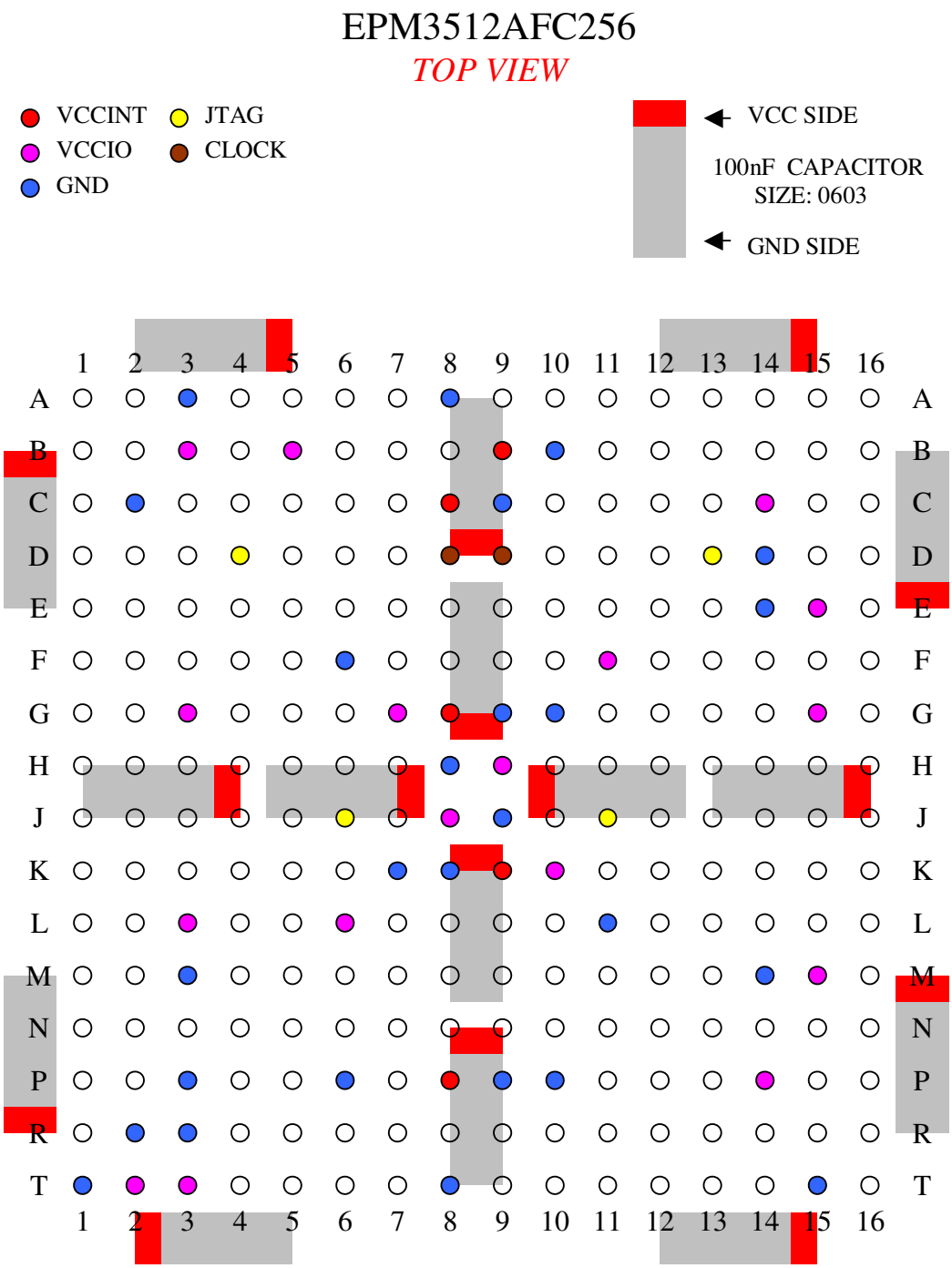

Figure 2.13.1 The FPGA EPM3512AFC256 decoupling on the LTU board

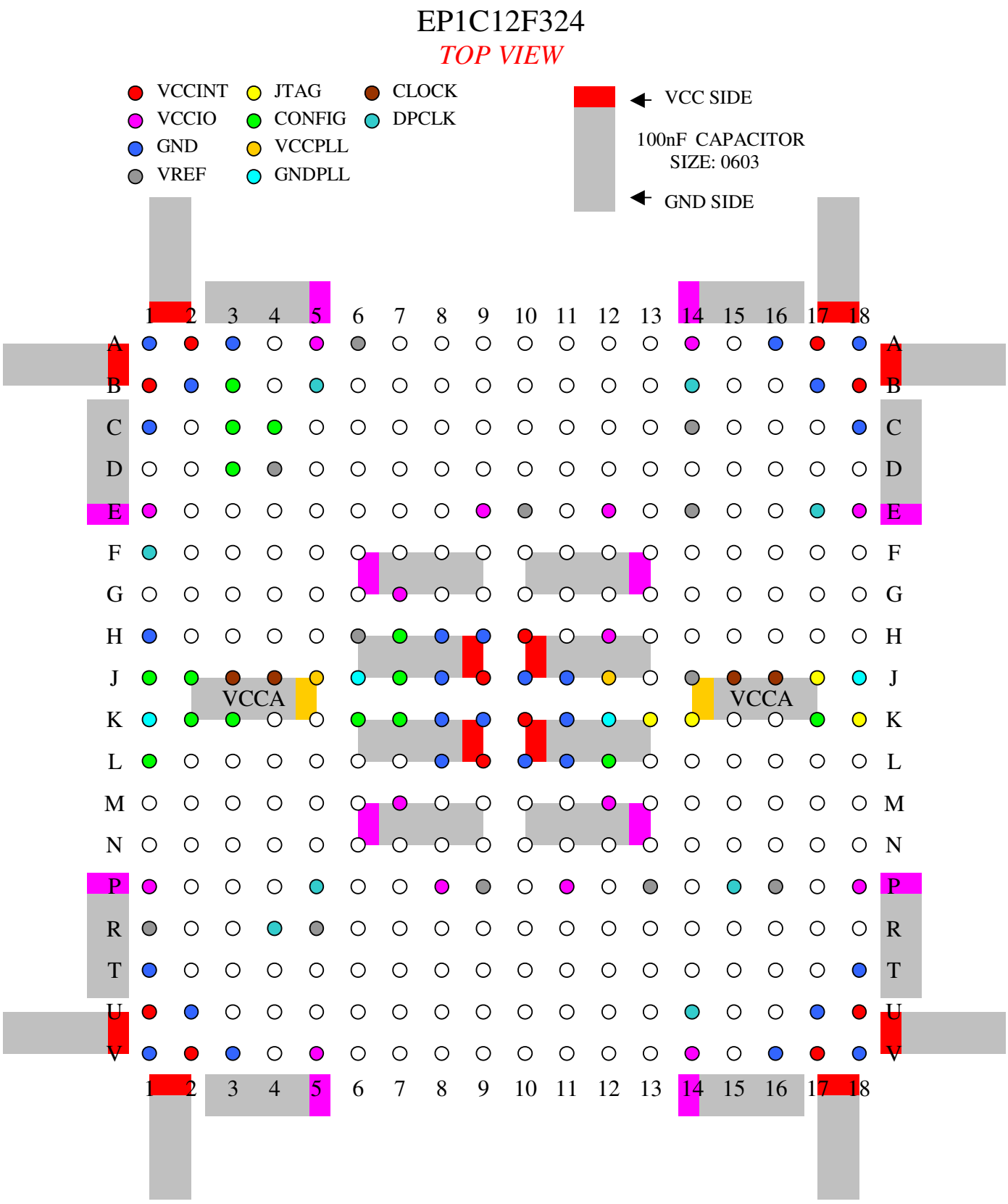

Figure 2.13.2 The FPGA EP1C12F324 decoupling on the LTU board

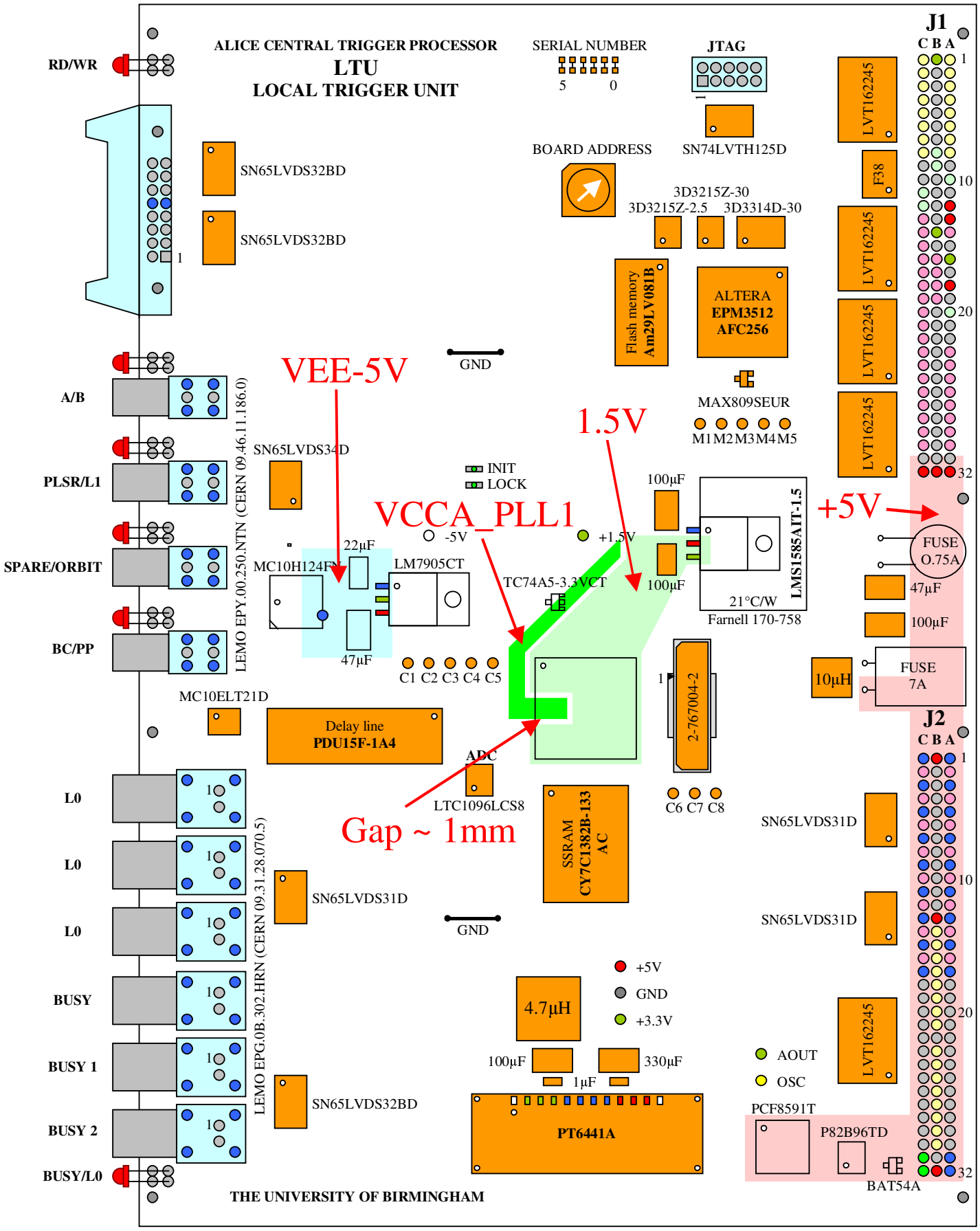

Figure 2.13.3 LTU board - +5V/+1.5V/-5V layer

2.15 TTCrx connection to the FPGA

Apart from the JTAG and the PROM connections, all the other TTCrx signals (58 in total) shall be connected to the FPGA.

The I^2C bus (SCL and SDA signals) enables the access to the TTCrx internal registers; the necessary interface and the control software could be developed at a later date. The SDA signal (open drain) requires an external pull-up resistor.

The TTCrx must be powered from the +3.3V supply; if +5V were to be used, the outputs would not be compatible with the FPGA input options.

2.16 Spare connections and test points.

For the possible future modifications (or, if need be, for the board patching), it would be prudent to include:

- around 4 spare point-to-point connections/tracks between the unused pins of the two FPGAs;
- around 4-8 “test points”, connected to the unused pins of each FPGA.

2.17 FPGA connections

This is just an attempt to *evaluate* the number of required connections to the two FPGAs; the correct figure shall emerge during the detailed design.

VME controller FPGA (208 I/O pins):

• VMEbus interface signals (A[23..1], AM[5..0], *etc*.)	38
• VMEbus data	8
• Internal VME signals (between the two FPGAs)	12
• Logic FPGA configuration signals	6
• Other signal between the two FPGAs (VME SYSCLK, Master Reset, spare, etc.)	6
• Flash memory interface	32
• Delay lines (optional)	8
• Board serial number	5
• Board address dial	4
• Other signals (power-on reset, test points, *etc*.)	6
• JTAG connections	4
Total	**129**

TTCit logic FPGA (249 I/O pins):

• VMEbus data	32
• Internal VME signals (between the two FPGAs)	12
• Logic FPGA configuration signals	6
• Other signal between the two FPGAs (VME SYSCLK, Master Reset, spare, etc.)	6
• LED connections	30
• SnapShot memory interface	41
• TTCrx connections	58
• L0 input, ScopeProbe outputs	3
• Test points	6
• JTAG connections	4
Total	**198**

2.18 TTCit board identification

The adopted scheme for the ALICE CTP board identification is described in [1], section **3.24**. In order to comply, the TTCit should provide the following data, set by the board hardware/firmware:

- *Board type code* (internal address 1): H”5A”
- *Board serial number* (internal address 2): 0 to 31 (5-bit code)
- *VME controller FPGA firmware version* (internal address 3): arbitrary 8-bit code
- *TTCit logic FPGA firmware version* (internal address H”20”): arbitrary 8-bit code