Mechanism Control Unit

Design Description

SPIRE-LAM-PRJ-000647

Prepared by :	Date :
Patrick Levacher Michel Jevaud Didier Ferrand	12/04/01
Checked by :	**Date :**
Dominique Pouliquen	12/04/01

Centre National de la Recherche Scientifique
Laboratoire Astrophysique de Marseille
BP 8 - 13376. Marseille Cedex 12.
Tel. : 33 .4.91.05.59.00 - Fax : 33 .4.91.66.18.62

Document change record

Issue - rev.	Date	Modifications
0 - 0	25-04-00	All : draft issue
1 - 0	19-06-00	First issue
1 - 1	25-12-00	Updated issue –all pages
2 - 0	12-04-01	All pages. LVDT device added. Schematics of MI1 model

Distribution list

Jean Louis Auguères	CEA	X	
Christophe Cara	CEA	X	
Colin Cunningham	ROE	X	
Brian Stobie	ROE	X	
Ian Pain	ROE	X	
Didier Ferrand	LAM	X	
Michel Jevaud	LAM	X	
Patrick Levacher	LAM	X	
Dominique Pouliquen	LAM	X	
Bruce Swinyard	RAL	X	
Ken King	RAL	X	

Acronyms

AD	Applicable Document
AVM	Avionics Model
BOL	Begin Of Life
BSM	Beam Steering Mirror
CQM	Cryogenic Qualification Model
EGSE	Electrical Ground Support Equipment
EOL	End of Life
ESA	European Space Agency
FIRST	Far Infra Red and Sub-millimeter Telescope
FM	Flight Model
FPU	Focal Plane Unit
FTS	Fourier Transform Spectrometer
FTSE	FTS warm Electronics
FTSP	FTS Preamplifier for the position encoder signals
H/K	House Keeping
H/W	Hardware
I/F	Interface
LAM	Laboratoire Astrophysique de Marseille
MAC	Multi Axes Controller
MCU	Mechanism Control Unit
N/A	Not Applicable
RAL	Rutherford Appleton Laboratory
RD	Reference Document
ROE	Royal Observatory of Edinburgh
S/C	Spacecraft
SM	Spare Model
SMEC	Spectrograph MEChanism
S/W	Software
TBC	To Be Confirmed
TBD	To Be Define
TBW	To Be Written
TC	Tele-Command
TM	TeleMetry
WE	Warm Electronics

Table of contents

List of figures

List of Tables

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to describe the electronics design for the control and monitoring of the HERSCHEL/SPIRE FTS Mechanism (**SMEC**) associated with the Beam Steering Mirror mechanism (**BSM**) control.
The control electronics unit of the 2 mechanical subsystems is called **MCU (Motorization Control Unit).**

The **MCU** is part of the **DRCU** and has electrical (Main Power Supplies) and mechanical (Electronics Cabinet) interfaces with it.

1.2 Applicable and reference documents

1.2.1 Applicable documents

AD1	Operating Modes for the SPIRE Instrument (SPIRE-RAL-DOC-000320)
AD2	FIRST/Planck Packet Structure Interface Control Document (SCI-PT-IF-07527)
AD3	Spire Spectrometer Mirror Mechanism Subsystem Specification (SPIRE-LAM-PRJ-000460)
AD4	FIRST / Planck Instrument Interface Document Part B (SCI-PT-IIDB/SPIRE-02124)
AD5	DRCU Electrical Interface Control Document (SAp-SPIRE-CCa-24-00)
AD6	SPIRE Instrument Requirements Specification (IRD) (SPIRE-RAL-PRJ-000034)

1.2.2 Reference documents

RD1	Beam Steering Mirror Control Software Requirements (Spire-ATC-Draft)
RD2	Beam Steering Mirror Warm Electronics (Spire-ATC-Draft)
RD3	Beam Steering Mirror Electronics Electrical Interface (Spire-ATC-Draft)
RD4	SPIRE Beam Steering Mirror Design Description v 3,4 (7 Feb 2001)

2 SYSTEM OVERVIEW

2.1 Main functions

The **MCU** is dedicated to the control and monitoring of the following **3 axis** of the SPIRE instrument:

- the Spectrometer Mechanism (**SMEC**). The control is typically based on a scan at a configurable speed, but can be set-up on the basis of a step position control in case on use of the step and integrate mode of the instrument.
- the **Chopper** and **Jiggle** axis of the Beam Steering Mirror subsystem . The control is a position step control pattern.

2.1.1 Axis control functions

The control of the 3 axis is performed by a 21020 DSP on the basis of trajectory generators and digital PID controllers associated with filtering for notching of mechanism modes.
The control parameters are put in memory for configuration purpose by mean of a command bi-directional 32 bit/330 kHz synchronous serial line connected with the DPU.

The software shall be based on a master scheduler on the principle of time sharing without the use of a specific multitask kernel. The tasks to be performed shall be called on a software interrupt generated by the inner DSP timer. The software interrupt defines the global sampling time (i.e. the computation cycle) of the DSP tasks @ a programmable rate between 100 us min and 300 us max[1] . At each cycle the following tasks are performed :

- the SMEC control loop task
- the chopper control loop task,
- the jiggle control loop task,
- the communication with the command line and other various internal DSP tasks.

2.1.2 Monitoring, H/K and telemetry function

The MCU shall provide 3 types of monitoring data for the 3 axis:

- **H/K,**
- **trace,**
- **telemetry.**

[1] The value of the main servo sampling rate shall depend on the final software load and functions to be performed. In the case of SMEC control, the arctangent computation and potentially the non-linear spline correction increase the software load to an amount which is not completely defined yet.

2.1.2.1 H/K Data

The HK data consist of the readout of single variable read by the command line @ a low rate (1 second typically and defined by the DPU). The H/K variables readout is made on the basis of a 'get_parameter' command with a reply within the DSP master scheduler time sampling rate added with the transmission delay. The complete overall delay is about 500 us.

Typically, the H/K variables may be all possible variables set-up or computed by the DSP, e.g.:

- mean SMEC scan speed
- SMEC control status
- SMEC encoder status
- Chopper mean position
- Jiggle mean position
- Etc ...

See the complete set of get_parameter commands to have the list of all available H/K variables

2.1.2.2 Trace data

The trace data is a buffer in the DSP data memory which is a long data acquisition. The number of trace variables, the sampling time, the length of the data buffer are programmable. This buffer can be read off-line, i.e. when the DPU sends a dedicated command for each data sample.

Typically, the trace data mode shall be used for the scanning of large number of samples @ high sampling rate for engineering purpose.

2.1.2.3 Telemetry data

The telemetry data are the SMEC time count between two encoder N micron positions (N is a configurable value between 2 and 26 microns by step of 2 microns) and BSM positions, sampled with a high level of synchronism with an external signal provided by the command line clock.

The telemetry data shall be transmitted on the fast 16 bits 1MHz synchronous serial line.

The telemetry line is independent of the DSP control and monitoring and is used for the delivery of data related to the detector signals , with high level of synchronism.

2.2 General architecture

The MCU control electronics includes:

- the **MAC Board**: common digital control board based on a **21020 DSP including:**
  - 3 DAC s,
  - 1 multiplexed 16 bits ADC for control,
  - 1 multiplexed ADC for telemetry,
  - 8 logical inputs
  - 12 logical outputs.

- **a SMEC Board**: analog electronics for the power amplification of the actuators and acquisition electronics for sensors preamplification and conditioning of the SMEC subsystem
- **a BSM Board**: analog electronics for the power amplification of the actuators and acquisition electronics for sensors preamplification and conditioning of the BSM subsystem

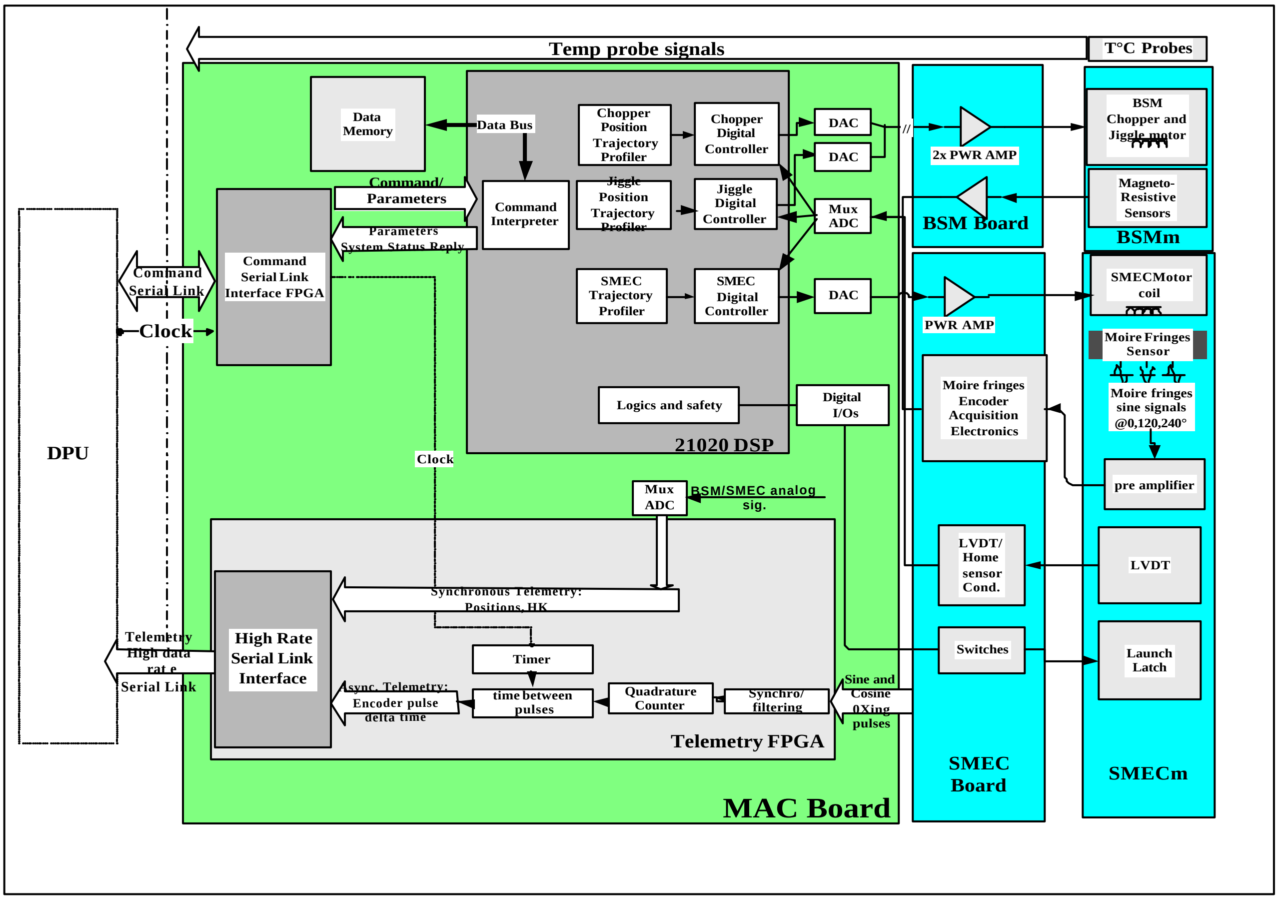


Figure 1: Control system architecture

2.3 Communication and telemetry principles

The MCU communication with DPU shall be done through 2 serial lines :

- Low rate command interface link @330 kbits, based on 32 Bits word.
- High rate telemetry interface link 1Mbits, based on 16 bits.

2.3.1 Low rate Command interface

This link is based on a Master/Slave relationship between DPU and MAC. By mean of a FPGA dedicated to serial interface, the commands and parameters are decoded and put in a register to be read by the 21020 DSP. There is a systematic reply to DPU for each command. For that, the DSP puts the dedicated parameter on the output FPGA register to be sent on the serial line. On the basis of this principle, the MCU DSP shall read the command register every cycle of the common time monitoring cycle of the 3 axis (i.e. 100-300 us).

The MCU shall receive from the DPU a 32 bits word for each command. The 32 bit word shall include (i) 2 bits for subsystem id (ii) 8 bits for command type (e.g. set scan_length), (iii) the rest for the parameter itself.

The MCU communication FPGA shall decode if the command concerns its subsystem, since all systems shall be addressed.
If yes, the MCU FPGA put the parameter value after header decoding in a register. The DSP reads periodically the interface register et put in its own memory the parameters according to a table pointer.

For every command, a handshake is done with DPU, so that only one command can be sent at one time until the parameter of the control shall be taken into account.

The handshake shall consist of only one 32 bits word return. Since the DSP readout of the command is done periodically every 100-300 µs, the maximum delay for a command to be taken into account is between about 200µs min and 500 µs max, randomly.

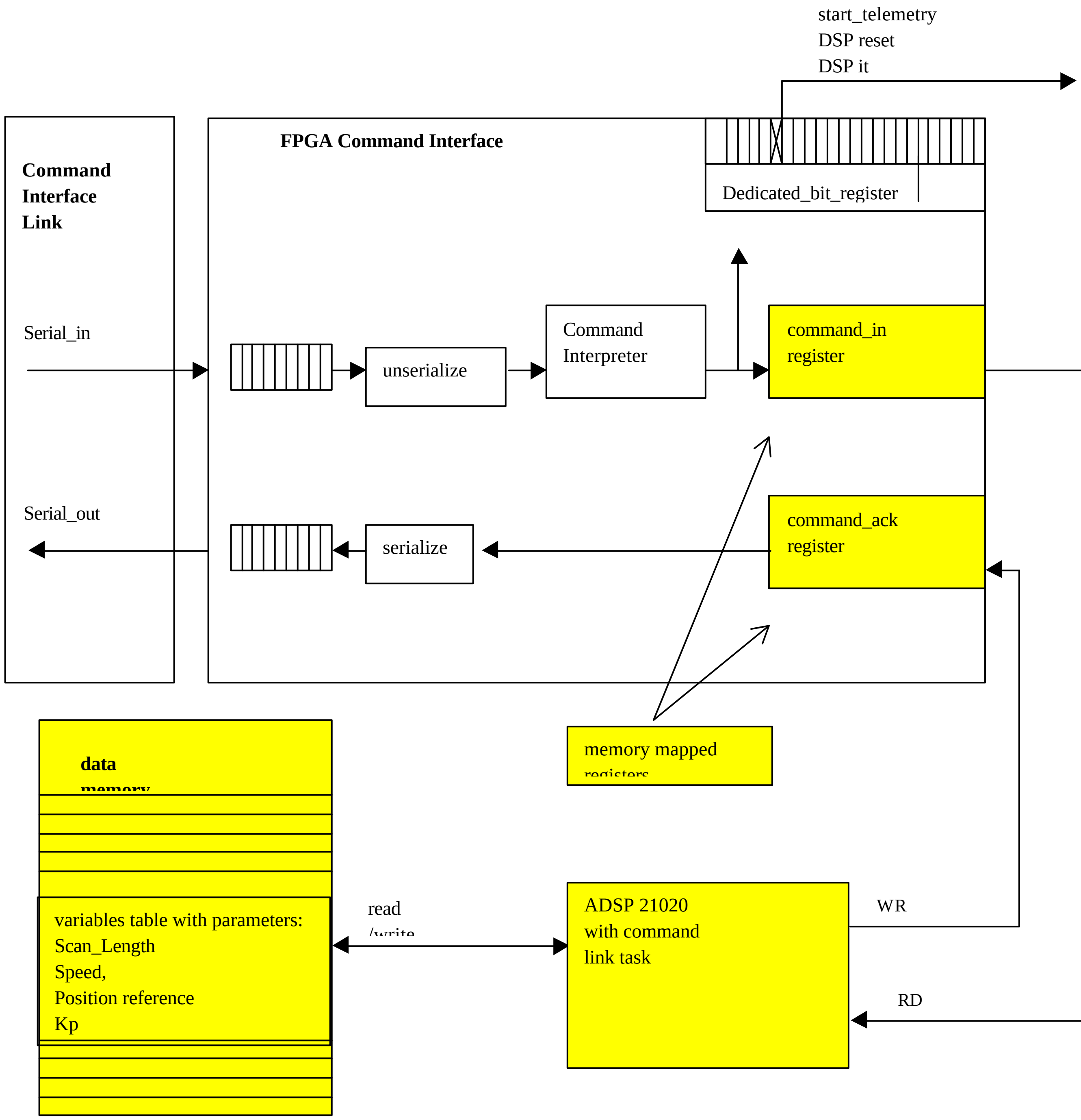


Figure 2: Command line and DSP link principles

2.3.2 High rate Telemetry serial link

The telemetry shall use a 1Mbit/s 16 bits serial link and is dedicated for data with high level of synchronism with detector data.

For the sharing of telemetry channel:

- there is only one common FPGA connected to one high rate serial link,

- this FPGA shall have two communication mode :

- **sync mode** : the telemetry data are the BSM axis with some additional SMEC information. In this mode the telemetry is synchronised by a clock

- **async mode** : the telemetry data are the delta time SMEC position with some optional BSM information. In this mode, the telemetry rate depends on the SMEC encoder pulses counts, i.e. the scan speed in an asynchronous way.

For the SMEC the telemetry packet as defined in the DRCU Electrical Interface Control Document (AD5), shall consists in a delta time between two zero crossing of a sine optical encoder signal, ie every 2 microns in the scan. At the nominal speed of 500 micron/sec, the delta time between two encoder pulses shall be delivered at a rate of 250 Hz.

The value of 2 micron is to be considered as a minimum value and shall be programmable. The possible values shall be from 2 to 26 microns every 2 microns step (TBC).

3 SMEC AXIS TRAJECTORY AND CONTROL

3.1 Scan in closed loop operation

The SMEC axis is controlled by a digital PID with a position ramp reference to insure a scan with configurable length and speed. The scan may be single with a fly-back at the end or a double scan (single and return) with the same common speed and length.

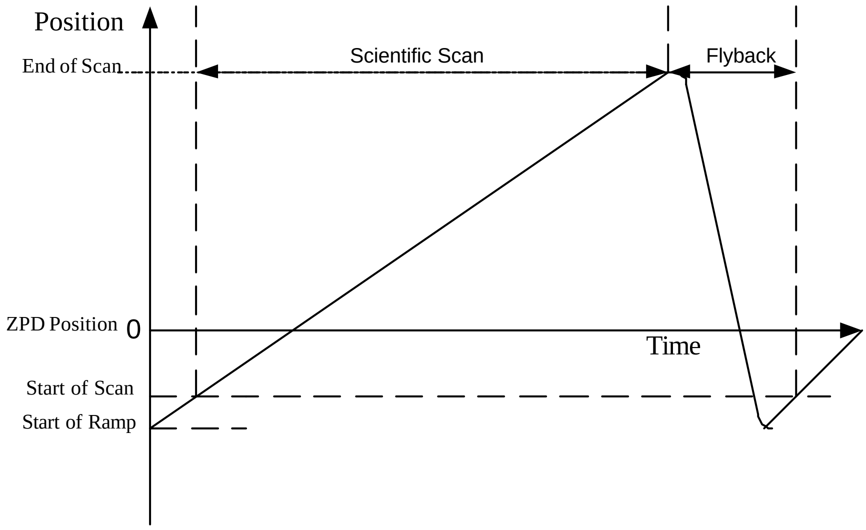


Figure 3: SMEC Scan characteristics

Scan characteristic	Specification/ (Initial Requirement)	Comment
Scan length	- 0.3 cm min to 3.2 cm max	Programmable length Value referred to the home position
Scientific Scan Length resolution	0.5 µm	based on a 16 bits parameter reference
Speed range	0 – 1mm/s	The value 0 means that the servo can be controlled in position steps.
Nominal scan speed	500 µm/s	The nominal speed is the basis for velocity stability requirements
Stability of the velocity scan	(10 µm/s rms).	on the basis of a filtered velocity fluctuation in the band 0.03-25 Hz
Fly-back duration	(less than 10% of the scientific scan)	including acceleration phase

Table 1 : SMEC scan specifications

3.1.1 Home position acquisition

The home position is delivered by the null differential signal provided by a 0.1 inch useful length LVDT.

3.1.2 Limits positions

There is no specific sensor for limits. A dedicated mechanical stop for metrology is foreseen on SMEC mechanism.
After the initialisation of the incremental encoder counter, the limits shall be defined by software.
Before initialisation of the incremental encoder counter, the limits can be found by a dedicated motion up to mechanical stop, or by the LVDT absolute position signal when the mechanism is within its range.

3.1.3 Optical Encoder Position Acquisition

The position is determined from the 3 sine optical encoder signals after pre-amplification, conditioning and a sine / cosine signals format conversion to be read directly by the DSP via a multiplexed 16 bits ADC for fine position on the basis of an arctangent computation in the DSP. The complete absolute position is then computed by addition of an incremental value of 2 µm every encoder cycle. Since that, the sampling time of the DSP does not depend on velocity (it is not the case with a 1/T speed counting between encoder pulses with some problems at very low speeds). Furthermore, the arc tangent computation of the position is dedicated for a fine and continuous position control, avoiding speed jitter that should be generated by encoder pulses if these signals enters directly in the control loop. The number of sine/cosine samples to be acquired to arctan computation is given by :

$$N = \text{Sine period} / (\text{Speed} * t_{sampling}), \text{ with } \text{Sine period} = 2\ \mu m$$

Note : by the mean time, the encoder pulses generated by the encoder electronics are entering an Encoder Unit Interface (FPGA) for pulses counting and crude position determination. The incremental position count is done in the FPGA for in order to verify the trajectory controlled by the DSP and acts the role of a watchdog. Furthermore, the Encoder Unit Interface includes error signals generation for whatever problem occurs during the trajectory (limits reached, etc...). If one on the logical signal is high, the DSP is interrupted and put in safe configuration.

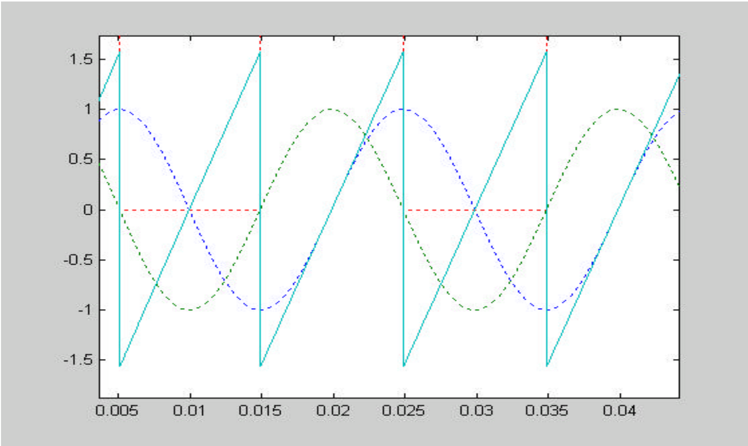


Figure 4: Fine position (solid line) derived by arctangent computation

3.1.4 Optical encoder LED control

Because of the possible decay of flux emitted by the optical encoder LED due to ageeing, 3 possible current levels can be set by the MAC board through the digital I/O port (0,low,high).

3.1.5 Degraded mode control using the LVDT

In case of optical encoder malfunction, the control loop may use the absolute position provided by the LVDT. The LVDT signals acquisition is done on the SMEC Board and digitalized by a 16 bits muxed ADC converter. The useful range of the LVDT , scaled on the basis of a quantization of 0.1 um, is +/- 3.27 mm.

3.1.6 Degraded mode control using the motor back emf

In case of optical malfunction, the control loop uses the LVDT for short range as described above. For a complete SMEC mechanism travel, the control shall work in a degraded mode using the motor back emf information and/or open loop parameters according to the mechanism stiffness law.

3.2 SMEC Position data for telemetry

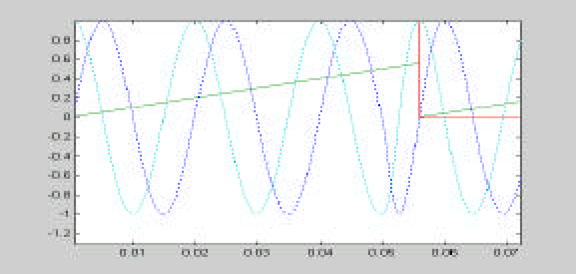
In order to reconstruct the interferometer signal, the detector readout data must be concatenated with position sampling data. There is no absolute time tag in the SMEC Control System, but an extra signal on the timer of the Telemetry Unit (FPGA) is implemented.

For this purpose, the SMEC control electronics provides the time elapsed between encoder pulses corresponding to, for example, 6 µm displacement (i.e.3 encoder engraving wave of 2 µm), by mean of a timer triggered by an external clock provided by the synchronous command serial line.

This is done by the Telemetry Unit on the basis of a zero crossing every 3 cycles (in this example, every N cycles more generally) as seen on the next figure. These data are then delivered to telemetry high data rate serial link.

The value of the number of encoder cycles is programmable with a minimum value of 2 micron corresponding to the zero crossing of an elementary sine wave of the moiré fringe optical encoder.

Figure 5: Time count (solid line) between encoder pulses for telemetry



3.2.1 Synchronization with DPU

The measurement of the delta time elapsed between 2 or more encoder zero crossing starts on the receipt of a specific synchronisation signal. This synchronisation signal resets the internal timer of the delta time count in the Telemetry FPGA. This function is implemented to allow synchronisation with detector data frames acquisitions. Since that the first delta time value provided to telemetry is representative of the time delay between the synchronisation signal provided by the DPU and the first optical encoder pulse of the scientific scan. The delta time counter is incremented by the Low Speed Serial link clock set to 330 kHz. The counter range is defined to 32 bits.

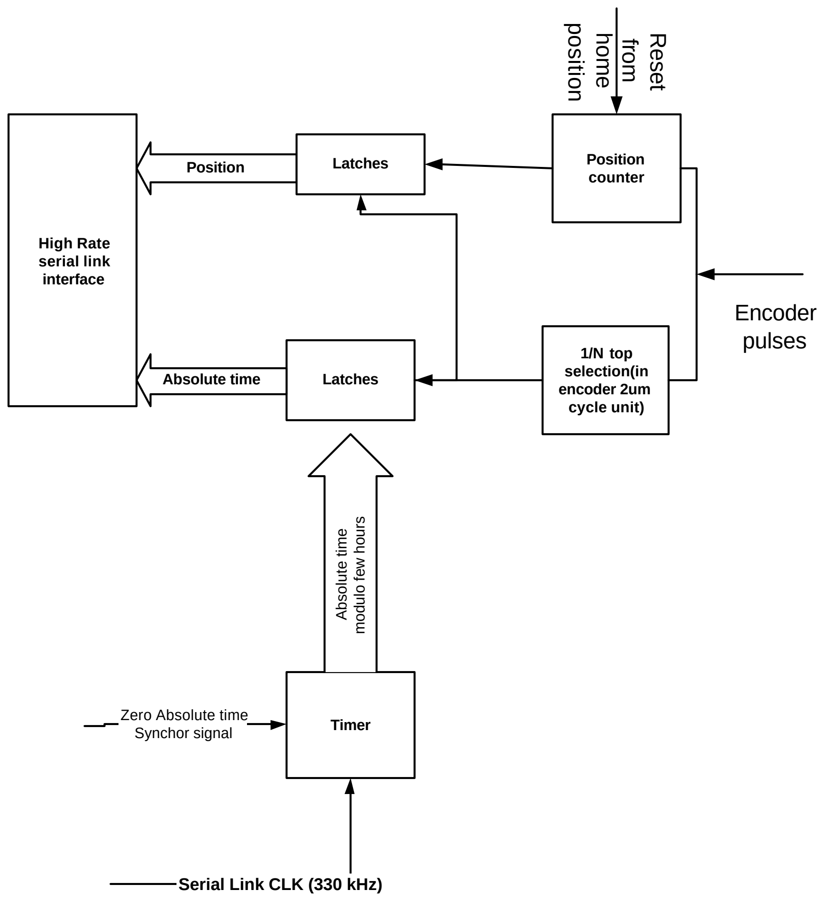


Figure 6 : Principle of position sampling timer synchronization with DRCU

Position data characteristic	As designed specification (Initial Requirement)	Comment
Position data sampling	2 µm (5 µm)	Provides a time count every encoder engraving of 2 µm Initial requirement based on the mini scientific wave length (200 um) divided by : - 4 (interferometer arm ratio) x - 5 (position data over sampling for calibration)
Position data accuracy	0.01 µm (0.1 µm)	Resolution given by the optical encoder after systematic non linearity correction
Position data synchronisation	time count start between pulse triggered by external signal	allows to reconstruct the trajectory starting from a reference start signal to be synchronised with start of detector samples

Table 2 : SMEC delta time positions specifications

3.2.2 Position data for telemetry sampling rate

The position for telemetry sampling data rate depends on the FTS SMEC velocity.
Consequently, i.e. the telemetry to be provided shall be asynchronous and not synchronised by a dedicated clock, the encoder pulses assuming to latch the delta-time counter.
The values of the sampling rate are proportional to the speed as specified in the following table. The sampling rate depends also proportionally on the length between the encoder pulses zero-crossing (e.g. every 2 or 4 sine waves).

Scan speed (mm/s)	Position data sampling rate (Hz) for 1micron delta time
200	100
500	250
1000	500

Table 3 : Telemetry rates for zero crossing every 2 microns on the optical encoder

4 PRINCIPLE OF OPERATION OF BSM CONTROL

The BSM is used to steer the optical beam of the SPIRE photometer channel over the detector arrays. The beam steering mirror will be used to perform jiggle mapping (to produce fully sampled images for feedhorn arrays, and to improve image quality) and perform chopping (to remove background and background variations and perform fine pointing corrections). This peaking up mode will be used to quickly ensure that point sources are well centred on individual detectors

The BSM comprises a flat mirror which is mounted on a two axis pivot system. This pivot system allows precise angular motion of the mirror over a small range of angular travel in two orthogonal axes. Electrical actuators are used to provide motion of the mirror. Magnetoresistive sensors are used to measure the mirror position to allow control of the mirror position. Control of each axis is done by a digital conventional PID (3-term) controller - with corrections for cross-coupling between axes- in the MAC Board.

4.1 control aspects (Abstract of ATC BEAM STEERING MIRROR CONTROL SOFTWARE REQUIREMENTS document)

4.1.1 INTRODUCTION

The BSM is controlled via software running on the DSP. The BSM software controls the position of the two BSM axes in response to external commands from the host software (DPU). Each axis can move independently.
The movement with respect to time is profiled via stored parameters to give a minimum energy, minimum noise position change, particularly for step commands.
In general the movements are repetitions of the same position/time profile.

In addition, in the event of measured behaviour resulting in a fault diagnosis, some system backup procedures are available.
Diagnosis of excessive position errors and analysis of recorded transient behaviour during operation can result in modifications to the control system by uploading different parameters into electrically-eraseable memory.

4.1.2 WAVEFORM COMMAND REQUIREMENTS

The BSM is slaved to the input demands at all times, so to perform a repetitive chop pattern the host processor has to issue a succession of position demands at the relevant times.

For example, to perform a 2 Hz chop between BSM positions p1 and p2, the following chop axis demand sequence and timing is required.
The command update rate (Ts) must always be >= 0.1 mS .

4.1.2.1 Time Demand

- 0.0 → (0.5-Ts) p1
- 0.5 → (1.0-Ts) p2
- 1.0 → (1.5-Ts) p1
- 1.5 → (2.0-Ts) p2
- 2.0 → (2.5-Ts) p1
- etc.

A step command waveform is assumed, and above a certain amplitude (10% of peak), it is profiled to produce a sinusoidal acceleration demand.
Other waveforms, such as triangular, can be approximated by a succession of incremental step demands, however the resolution will always be dependant on the update rate.

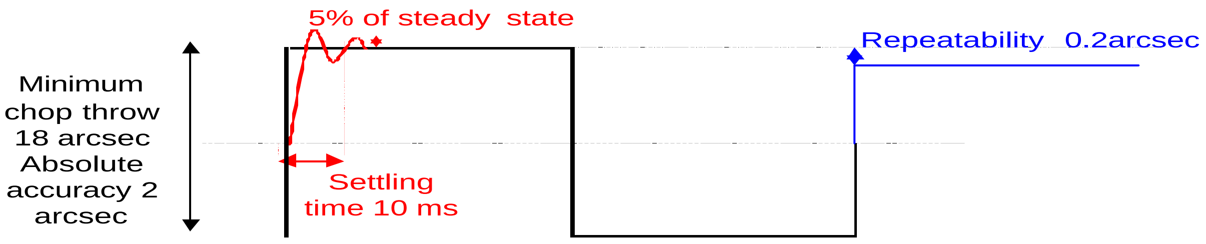


Figure 7: BSM Trajectory specification

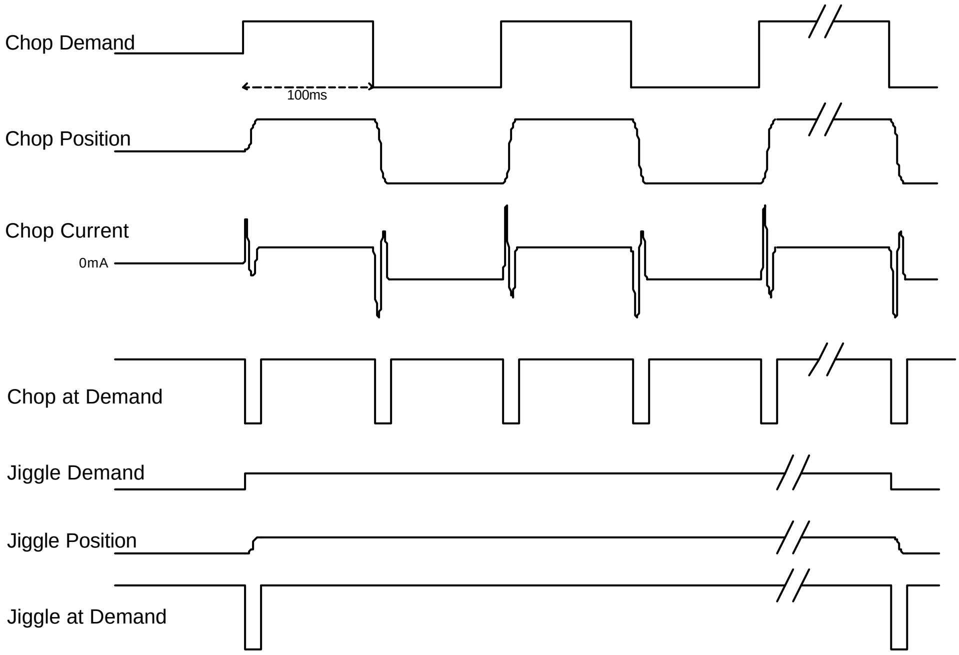


Figure 8 : BSM Control system chronogram

4.2 Telemetry data list

The telemetry data shall be the Chop and Jiggle axis positions @ 20 ms rate (TBC).

5 MAC OPERATION

The MAC Board may be operated by a set of commands according to the following table.

Mode #	Mode Name	Action	Possible Next Mode	On Command
0	Off	The set of boards is not powered. No function available	On	Power On
1	On	The set of boards (main or redundant) board is powered .	Off	Power Off
1.0	Booting	• The software is initialized after reset. • The software is downloaded from Prom memory to SRam memory	Downloaded	MAC Reset On
1.1	Downloaded	• The communication is available. • The status of the system is available. • Waiting for commands • Goes initially to Open loop mode	Booting	MACReset On **'ResetMAC'**
1.1.0	Open Loop	• Waiting for Advanced Configuration Commands (Control gains,etc ...) • Goes initially to open loop standby	Close Loop	'**CloseLoop**'
1.1.1	Close Loop	• nominal control loop mode using a digital PID controller at 300 µs sampling rate • Goes initially to Stand By mode	Open Loop	**'OpenLoop'**
1.1.1.0	CLoop Stand By	Waiting for motion configuration: (Scan Length, etc..)	Motion init	**'Init Motion'**
1.1.1.1	CLoop Motion Init	Searching for limits and diagnostics	Ready	End of Motion initialization
1.1.1.2	CLoop Ready	Waiting for motion commands and configuration	Scan	'**StartMotion'**
1.1.1.3	CLoop Scan	Motion underway: scan for the SMEC, chopping and jiggling for BSM	Ready	End of Motion '**StopMotion**' **'ResetMotion**

5.1.1 Command types

There are typically two types of commands :

Set Command : the parameter is put in the DSP memory for immediate or delayed execution
Get Command : the dedicated parameter in memory can be read by the DPU

5.1.2 Bufferized commands and parameters

Some parameters set by a Set Command should not take immediate effect on the control. These parameters are called **bufferized parameters**. They are taken into account when a specific **Update** command is sent. This is useful for a configuration set definition such as the scan speed and length before a validation of the configuration. The parameters that are bufferized are :

- ClearPositionError,
- Acceleration,
- deceleration,
- Position,
- Velocity,
- Scan length
- PID parameters,
- MotorCommand.
- Etc ...

5.1.3 List of parameters associated with the Set/Get Command

The complete list of parameters associated with the Set or Get command is listed after. The parameter numbering is up to now given as examples and may change during development until the first software issue.

5.1.3.1 Digital Servo Filter parameters

#	Parameter	Action
1	**Kp**	Proportional gain of the digital PID controller
2	**Kd**	Derivative Gain of the digital PID controller
3	**DerivativeFilter**	Sets the filtering time constant to calculate the derivative term.
4	**Derivative**	Returns the derivative of the current position error as calculated by the servo filter. The derivative value is defined as the current position error minus the previous position error.
5	**Ki**	Integral gain of the digital PID controller
6	**Integral**	Returns the current integrated position error of the servo may be used for Integration limit survey
7	**IntegrationLimit**	Loads/Reads the integration saturation for the integral compensation of the servo
8	**PositionErrorLimit**	determine/reads the position error value that causes an error on the servo
9	**PositionError**	In case of the position error exceeds a threshold (**PositionErrorLimit)** , the motion is stopped for further diagnostic. This parameter is used to reset the position error by forcing the position reference to be equal to the actual position counter. With a get command, Returns the difference between the actual position and commanded position
10	**MotorBias**	Sets a bias voltage of the digital servo. Used for a static effort cancel out on the mechanism
11	**MotorLimit**	Sets a saturation on the output of the motor output command issued by the servo filter
12	**SampleTime**	Sets the cycle time for the servo
13	**NotchParam**	Parameters for notch filtering of mechanical modes. Here is the pole compensation of the first parasitic mechanical mode

5.1.4 Trajectory Profile commands

#	Parameter	Action
14	**Position target**	specifies the trajectory destination For BSM= position step level (um) For the SMEC : moves the SMEC by a step by step way. Unit= 1 um Range: 5.5 cm: 55000 values B0-B21: position reference from the home position With Get Command, allows the measurement of a position step by step. Usefull for engineering mode to get position for step response. B0-B21: actual position from the home position
15	**CommandedAcceleration**	returns the current commanded acceleration value for the specified axis.
16	**CommandedPosition**	The commanded position is the instantaneous position value output by the trajectory generator
17	**CommandedVelocity**	The commanded velocity is the instantaneous velocity value output by the trajectory generator.
18	**Acceleration**	Load the max acceleration of the motion
19	**ActualPosition**	loads the actual position register and at the same time the commanded position
20	**Jerk**	Loads the acceleration variation limitation on the trajectory of the specified axis
21	**ScanLength**	Load / update a table defining a spectrometer scan. The scan is defined as a start position, i.e. a number of position sensor fringes move.
22	**ScanSpeedRef**	Load/Read the desired speed of the control scan reference speed: 16 bits if value=0 the servo is in position mode
23	**ProfileMode Axis i**	**'SetProfileMode'=0 =>** Mode= **Position_Closed_Loop** **'SetProfileMode'=1** => Mode= **Scan_Closed_Loop** The number of scan to be performed is within parameter

5.1.5 I/O and Encoder signals related commands

24	**CaptureValue**	Returns the contents of the Position Capture register
25	**ActualPosition**	reads the contents of the encoder's actual position counter
26	**CaptureSource**	Defines which signal is used for encoder counter initialisation (Home, Mechanical Limits ,Current position)
27	**EncoderSource**	Set the type of encoder feedback. Can be incremental mode using sine cosine A/B signals, or arctangent computation
28	**EncoderRatio**	Converts the number of encoder pulses in scan length micron units
29	**SignalStatus**	Returns the signal status register: Encoder A, Encoder B, Home , Positive limit, Negative Limit
30	**Limits Mode**	Enables or disables the limits sensing

5.1.6 DSP Software related commands

31	**Time**	returns the number of cycles since last reset
32	**Version**	Returns software version number
33	**Reset**	restores the chipset to initial default conditions
34	**ResetEventStatus**	reset the event status and related error bits for re-initialisation
35	**Update**	Updates in the chipset the buffered parameters (e.g. PID parameters)

5.1.7 Communication related commands

36	**GetHostIOError**	returns the IO Error code
37	**IOPollingTime**	Set a maximum time between 2 commands before a IO Error code is generated, meaning a communication problem
38	**NoOperation**	No effect on the chipset.It is usefull as a 'null' operation to verify communication with the Motion Processor
39	**PortMode**	Determines the instructions set that can be executed. When set to Limited, only the base instructions may be used

5.1.8 Motion and control modes related commands

#	Parameter	Action
40	**EventStatus**	reads the event register for a specified axis.
41	**ActivityStatus**	reads the Activity Status Register. This Status Register can only be read and no bit can be cleared since the word is refreshed by the chipset.
42	**CurrentMotorCommand**	Returns the current motor command for the specified axis. In closed loop, this is the output of the servo filter. In the open loop mode it is the contents of the motor output command register.
43	**MotionCompleteMode**	establishes the source for the comparison which determines the motion complete status
44	**MotorMode**	**'SetMotorMode'= 1** => Mode = **Closed_Loop** **'SetMotorMode'= 0** => Mode = **Open_Loop** **with get command :** The error / internal mode status word allows the DPU to identify problems B0-B3: idem set_SMEC_mode B4-B21: error/internal mode status
45	**SettleTime**	Sets the time that the specified axis must remain within the tracking before the axis settled indicator is set to 1
46	**SetAutoStopMode**	Defines the conditions on which the motion should stop. Stops the specified axis in abrupt or smooth mode
47	**Breakpoint**	Defines an action of a specified event

5.1.9 Trace and telemetry related commands

The trace mode allows to record up to N data in a memory allocation for further read out of servo loop values. The trace table is , after recording, read by a dedicated command on the low rate serial command line (point to point).

#	Parameter	Action
48	**TraceCount**	Returns the number of points stored in the DSP Trace Buffer it defines the length of telemetry sample
49	**TraceStatus**	B0= 1 when in synchronous mode, 0 when in asynchronous encoder pulses mode
50	**TraceMode**	The trace may be in rolling on or one shot time
51	**TracePeriod**	Sets the time period, expressed in ms, between successive trace points. or the number of encoder zero crossing to be sampled in asynchronous mode
52	**TraceStart**	0=Immediate start of the trace 1= Stop on next Update command 2= Event Status Register + mask of the register for trace trigger
53	**TraceStop**	0=Immediate stop of the trace 1= Stop on next Update command 2= Event Status Register
54	**TraceVariable**	Specifies up to 4 variable to be put in the DSP trace register and sent to telemetry channel. It consists in the telemetry data muxed selection

55	**TelemetryCounterReset (broadcast)**	Starts the telemetry counter on a broadcast synchronised signal

5.1.10 HK on the command line related commands

This commands allow to recover some data for H/K in order to build a table representative of the inner state of the servo control.

#	Variable	Action
56	**ActualVelocity**	Get the instantaneous velocity
57	**ScanMeanSpeed**	Actual scan mean measured speed Allows to verify the velocity scan error
58	**MeanSpeedMeasure**	Length of the speed sample used for Mean Speed measurement
59	**ReadAnalog**	Returns a 16 bits value representing the value of a specified analog input
60	**ReadLogics**	Returns a 8 bit value of the logical inputs of the MAC
61	**ReadMemory**	Returns the value of whatever DSP memory mapping

To be continued ...

5.1.11 Event Status Register

There is one event register per axis.

Bit	Name	Description
0	**Motion Complete**	Set when a trajectory profile/scan is complete
1	**Position wrap**	Set when the actual motor position exceeds 7FFF.. (the most positive position) and wraps to 8000... (the most negative position)or vice versa
2		
3		
4	**Motion error**	Set when the actual position differs from the commanded position by an amount more than the specified maximum position error
5	**Positive limit**	Set to 1 when the positive limit switch is active
6	**Negative limit**	Set to 1 when the negative limit switch is active
7	**Instruction error**	Set when an instruction error occurs
8		
9	**@ Maximum velocity**	Set to 1 when the trajectory is @ maximum velocity. This bit is determined by the trajectory generator, not by the actual encoder position.
10	**Tracking**	Set to 1 when the axis is within the tracking window
11	**Current profile mode**	on 3 bits: velocity scan, position step, trapezoidal
12	**Axis settled**	
13	**Motor Mode on/off**	
14	**Position capture**	
15	**In-motion**	

5.1.12 I/O Error code register

The Error code register is a 16 bit word encoding the error that occurs in the MAC board. In normal cases the command line replies to DPU with the same input command as acknowledge. If there is a problem, the reply shall consist in the following error code register word.

Error #	Name	Description
0	**No error**	
1	**Processor reset**	
2	**Invalid instruction**	
3	**Invalid axis**	
4	**Invalid parameter**	
5	**Block out of bounds**	
6	**Bad serial CKS**	
7	**Not primary port**	
8	**Invalid parameter change**	
9	**Invalid move within limits**	
10	**Time out of Command link FPGA**	The command line FPGA sends to DPU the error status word with this bit set to one if the DSP does not read the command within a time out.
11	**Time out of Command Maximum Delay**	
Etc	**To Be completed**	

5.2 Command format

A command is a 32 bits word with the following allocation:

Bits 29-28	**Bits 27-20**	**Bits 19-18**	**Bits 17-0**
MCU header	Command type	Axis id.	Parameter

Axis id:
0: all axis
1: SMEC axis
2: Chopper axis
3: Jiggle axis

Examples:

- Load the speed for the SMEC:
  - MCU header: b10
  - Command type: set=1 ScanSpeedRef=d22 =h16
  - Axis id= 1 (SMEC)
  - Parameter= 500 micron/s =h1F4

=> Complete 32 bit word: 00 10 1 0001 0110 01 0 0000 0001 1111 0100 = h 28 B2 01 F4

- Update the parameters for the SMEC

- MCU header: b10
- Command type: set=1 Update=d35 =h23
- Axis id= 01 (SMEC)
- Parameter= 0 (NA)

=> Complete 32 bit word: 00 10 1 0010 0011 01 0 0000 0000 0000 0000 = h 29 1A 00 00

5.3 Typical commanding scenarios

5.3.1 Typical nominal FTS commanding Scenario

The first scenario is a typical commanding scenario for the FTS control

Time	Command sent on command link	Reply	Mnemonic	Action
When DPU wants to configure the speed	28 B2 01 F4		SET ScanSpeedRef	Puts the 500 microns/sec reference speed in memory
0.5ms after		28 B2 01 F4		Reply of the same word to DPU as acknowledge
When DPU wants to start a scan	29 1A 00 00		UPDATE	Start the scan with the parameters in the memory
0.5ms after		29 1A 00 00		Reply of the same word to DPU as acknowledge
When DPU wants to know the status of the scan			GET Event Status register	If the scan is not finished, the first bit (Motion complete) is =0, else=1
1 s after (up to DPU)			GET Mean Speed	Reads the last calculated mean speed for H/K
1 s after (up to DPU)			GET Event Status register	If the scan is not finished, the first bit (Motion complete) is =0, else=1
Etc ...				

5.3.2 Initialisation commanding Scenario

To be Done

5.3.3 Trace Acquisition commanding Scenario

To Be done

6 MAC SOFTWARE

6.1 21020 DSP Software principles

The DSP software is in 21020 assembly language without the use of a specific off-the-shelf real time operating kernel. The assembly language is chosen because Analog Devices provides directly specific libraries to produce PID, filtering, arctangent computation with a high efficiency and readability. For this reason C coding is not foreseen since the software shall be simple, dedicated to pure signal processing (ie realisation of IIR Filters with product/summation operations) without complex command interpreting nor complex mode of operations.
Code uploading in the DSP is not foreseen, the DSP being only used for control, in substitution of analogue electronics, with very poor load regarding communication / command interpreter. Code uploading would imply a very heavy software management task.

6.1.1 ITs

The main tasks to be performed are called by a Master scheduler which is a routine interrupted by a software interrupt generated by the inner timer of the DSP.
Mainly, the software does not use other interrupt, excepted for the following functions :

- watchdog interrupt to recover from a loss of control of the DSP,
- interrupt on specific synchronisation commands between the DPU and the DSP.
- interrupt on each SMEC incremental encoder pulse for position counting.

6.1.2 Watchdog mechanism

A Watchdog mechanism is implemented between Control Bus FPGA and DSP to monitor the DSP behaviour. This counter is incremented on the basis of the Control Bus FPGA internal clock.
If the DSP does not ensure the counter reset, the FPGA automatically resets the DSP.

6.1.3 Reset

The reset of the DSP shall be activated on the following events :
- no watchdog signal
- specific command line word: a time out in the command line FPGA, triggered on the arrival of a new command and reset by the DSP read-out, shall detect a potential loss of the DSP

6.1.4 Motion in safe configuration

In case of problem detected by the control (position error greater that a threshold value, loss of encoder pulses, etc ...), the software shall immediately open the axis control loop and force dac to 0 volt. A redundancy in the fault condition shall be implemented to detect possible memory latch-up. In case of ram memory partial destruction, an other mapping may be specified.

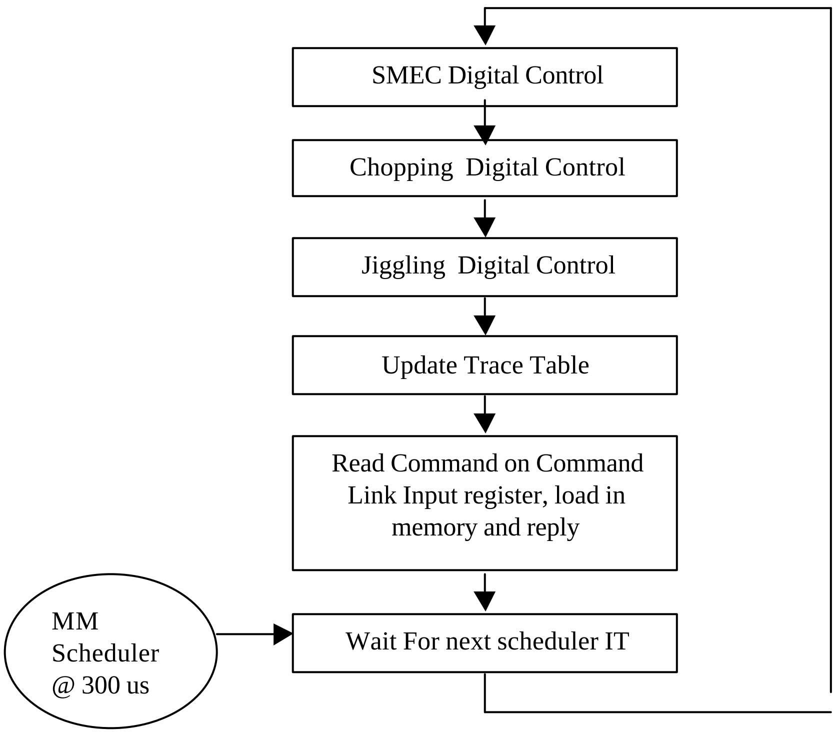
6.2 MAC Software architecture

The software is designed to be as simple as possible. For this purpose, there is an only task running every time defined by the digital servo sampling time parameter. The digital servo sampling time is defined by the MMS (Mac Master Scheduler).

6.2.1 MAC Master Scheduler (MMS)

The principle of the MMS is the management of the main cycle to be performed in the MAC DSP.
The main cycle is set by default to 300 us (TBC), but is configurable.
The MMS uses the internal clock of the DSP , incrementing an internal counter set to deliver an interrupt every main cycle interrupt signal triggering the start of the main task.

6.2.2 MAC Main task



6.2.3 SMEC and BSM axis Control Algorithms

- Read Digital I/O (Home, Launch Lock)
- Compute control status

- Compute servo loop

6.2.4 SMEC Servo Loop algorithm

- Step 1 : read sinus and cosinus (1 DSP cycle = 50 ns)
- Step 2 : read cosinus (1 DSP cycle = 50 ns)
- Step 3 : filtering sinus (T1 = 40 DSP cycles = 2 µs)
- Step 4 : filtering cosinus (T1=40 DSP cycles = 2 µs)
- Step 5 : compute arctangent (T2 = 120 DSP cycles = 6 µs)
- Step 6: compute current trajectory,that is the same that a speed filtering (T1=40 DSP cycles=2 µs)
- Step 7 : compute current error and "do something" if error > max error (T3=40 DSP cycles=2 µs)
- Step 8 : with error, compute output DAC for FTS motor amplifier (T4 = 80 DSP cycles = 4 µs)
- Step 9 : write output DAC (1 DSP cycle = 50 ns)

T1, T2 are roughly given and are to be exactly defined.
T3,T4 and T6 are estimated with Analog Device application notes, for example the number of cycles for a FIR Filter is given by : **number of cycles = 7 + number of taps used for filtering.**
Another example is the compute of **PID in 10 cycles** (ref : DSP-Based Motor Controller Seminar of Analog Devices)
These numbers don't include error test , overflow… so we must reserve about 20 cycles for this.
The compute of **Arctangent** is given for **82 cycles** maximum from Analog Devices (ref ADSP-21000 Family Application Handbook), but we must reserve about 40 cycles for overflow and error tests.
So, the total timing for execute step 1 to 11 can be estimated to about 17 µs.

6.2.5 BSM Chopping axis Servo loop

Nota : the BSM software is specified in the Reference Document RD1 Beam Steering Mirror Control Software Requirements (Spire-ATC-Draft)

- Step 1: set analog multiplexer to BSM chop position magneto-resistive sensor (1 DSP cycle = 50ns)
- Step 2: wait 300 ns (analog mux transition time)
- Step 3: write Start of Conversion to ADC (1 DSP cycle = 50 ns)
- Step 4: wait 15 µs (ADC conversion time)
- Step 5: read position (1 DSP cycle = 50 ns)
- Step 6: compute position filter (T1 = 40 DSP cycles = 2 µs)
- Step 7: compute current error and "do something" if error>max error (T2 = 40 DSP cycles = 2 µs)
- Step 8: with error, compute output DAC for BSM Chopping axis (T3 = 40 DSP cycles = 2 µs)
- Step 9: write output DAC (1 DSP cycle = 50 ns)

So, the total timing for execute step 1 to 9 can be estimated to about 7 µs.
The steps are exactly similar to the previous task.

7 REDUNDANCY AND DEGRADED MODES

The design is oriented in order to avoid :

- single point failures by mean of dedicated redundancy,
- propagation of failures to other sub systems by mean of specific protections.

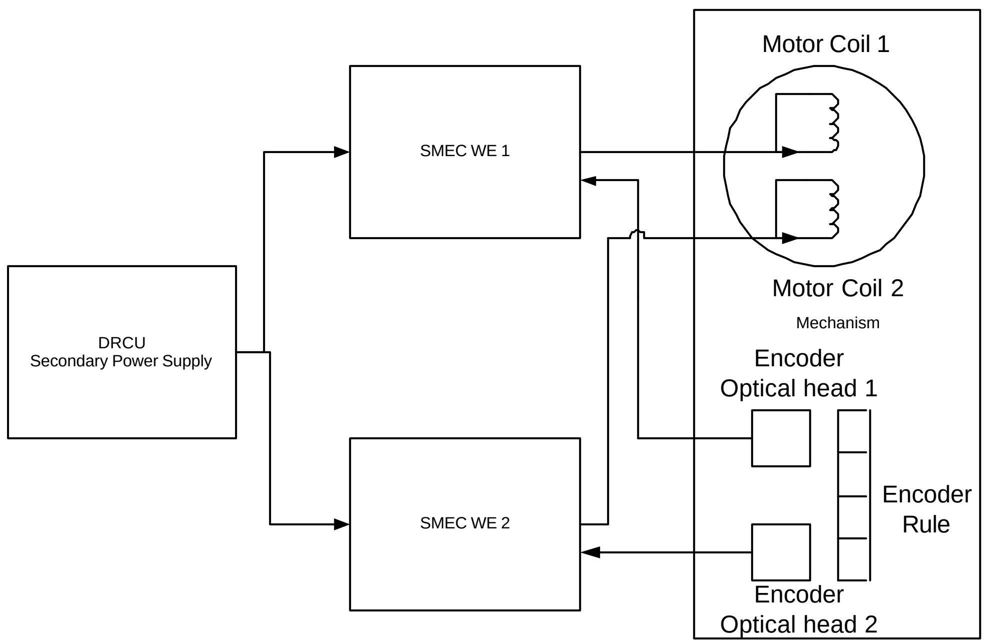
7.1 Redundancy

The SMEC Control electronics consists of 2 independent sets of boards without cross switching.
The related reliability is expressed by :

WE1.M1.OE1 + WE2.M2.OE2

Wei : FTS electronics board
Mi : motor coil
Oei : optical encoder

Note : the solution with cross switching of the components would imply non reliable complexity and single point failure with the following reliability layout : ***(WE1+WE2).(M1+M2).(OE1+OE2).Ics with*** Ics : Necessary Interface for cross switching



7.2 Limitation of failure propagation

The limitation of the failure propagation is ensured by design as demonstrated in the following AMDEC analysis :

Subsystem/Component in failure	Event during a specific mode of operation	identification mean	Effect	Recovering Degraded mode
Interface electronics	No command can be received	the DPU has no command handshake	FTS WE out of control	
	Bad command/parameter	Checksum/No recognition of a command		Stop of the scan Status report Switch to stand-by mode waiting for another control mode operation
DSP and related interface components	Control algorithm diverging	software limits	Increase of the actuator command up to limitation	Stop of the scan Status report Switch to stand-by mode waiting for another control mode operation
	No control on the DSP	Watchdog		Emergency stop on watchdog action
Motor and related PWR amp	Short cut in the motor coil	Motor current measurement	Limitation of current value by the PWR amplifier	Use of the redundant board + motor coil.
Optical encoder and related acquisition electronics	Loss of some individual incremental pulses during closed loop operation	Difference between fiducial mark and position counter	Perturbation on the trajectory	Status report Adjust the position counter on next scan
	Total loss of incremental encoder pulses	id.	Error vs ramp reference increase -> Increase of the actuator command due to integral effect and position ramp increase -> Excessive motion speed up to speed limitation. -> Emergency stop.	Stop of the scan Status report Switch to stand-by mode waiting for another control mode operation
	Glitch on the absolute position counter (decreasing)	Additional position switch on the mechanics	The scan goes to hardware limits	Flyback operation and reset of the absolute position counter
	Glitch on the absolute position counter (increasing)	The scan ends before the nominal scan length parameter, and a difference between fiducial mark and position counter occurs at the end of flyback	The scientific scan is not achieved completely	Status report Readjust the position counter on next scan

7.3 BSM Redundancy and degraded modes

- SPIRE BSM will have 2 position sensors per axis, all permanently connected to separate analogue inputs,
- The motors will wired and driven so that failure of one coil allows operation at reduced performance

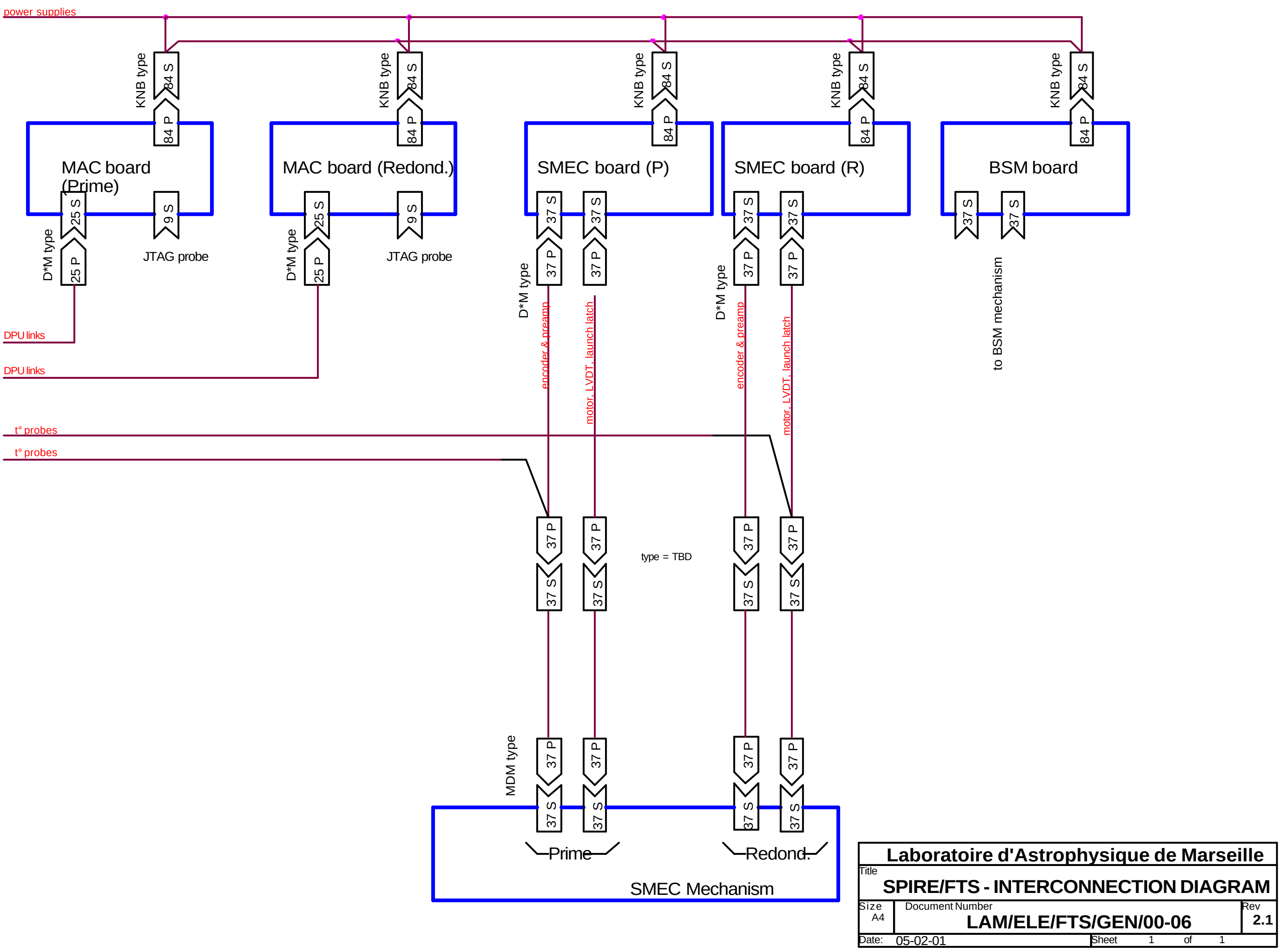
The BSM redundancy is described in the following document RD4: SPIRE Beam Steering Mirror Design Description v 3,4 (7 Feb 2001)

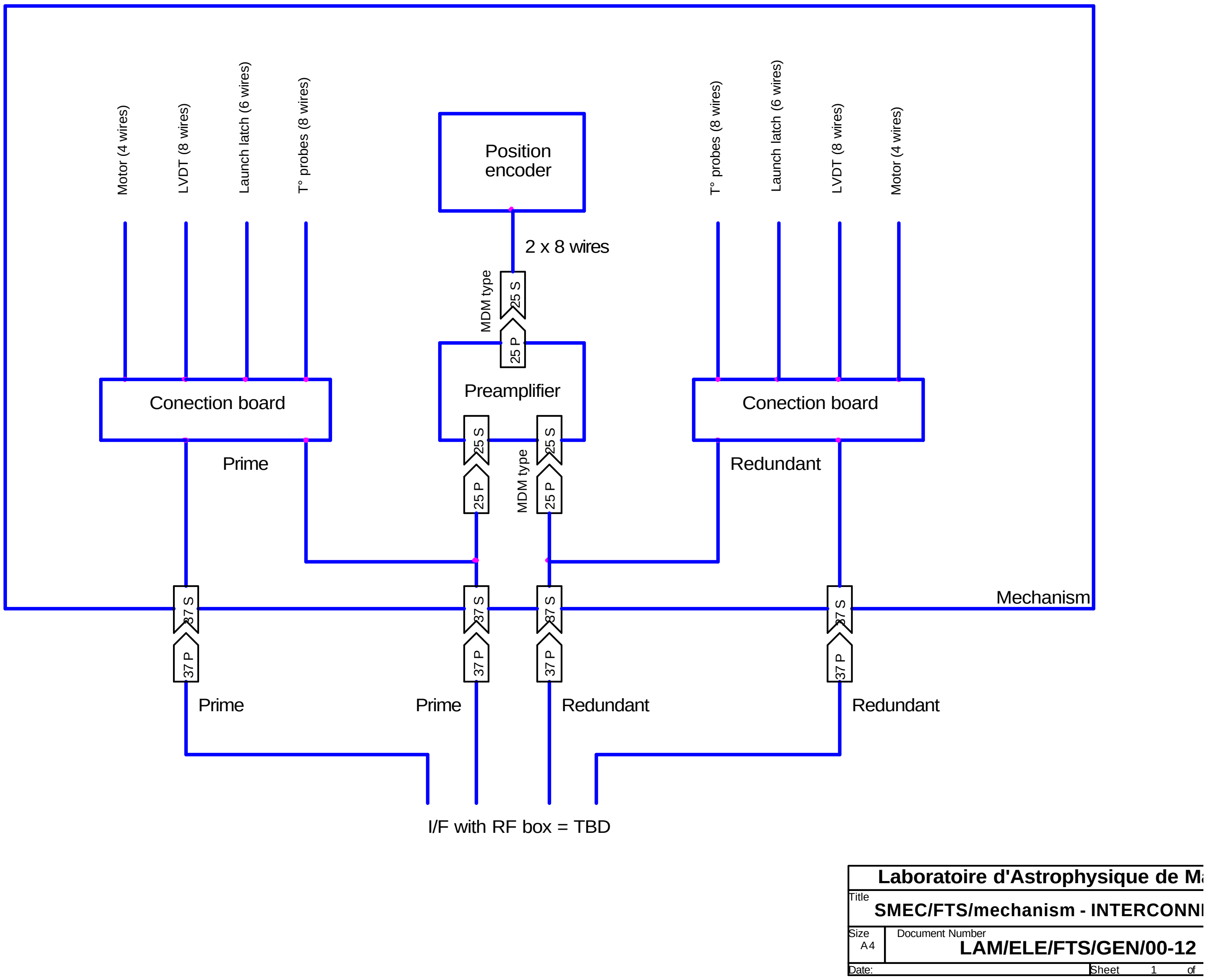
Failure Mode	Effect	Remedy	Criticality
Chop Pos Sensor 1 fails	Cannot measure Ch1 position	Use C2 pos coil	No effect
Chop Pos Sensor 2 fails	Cannot measure Ch2 position	Use C1 pos coil	No effect
Chop Drive Coil 1 fails off	Cannot move from nominal position	Use C2 drive coil on its own	Longer rise-time – lose some efficiency
Chop Drive Coil 2 fails off	Cannot move from nominal position	Use C1 drive coil on its own	Longer rise-time – lose some efficiency
Chop Drive Coil 1 fails on	Cannot move from last position	Switch off BSM controller	Cannot chop or jiggle – only scan mode
Chop Drive Coil 2 fails on	Cannot move from last position	Switch off BSM controller	Cannot chop or jiggle – only scan mode
Both Pos Sensors fail	Cannot measure position	Run open loop	Reduced accuracy of chop or jiggle – or use scan mode
Both Drive coils fail off	Cannot move from nominal position	None – 2 point failure	Cannot chop or jiggle – only scan mode
Mechanism sticks	Cannot move from last position	None	Cannot chop or jiggle – only scan mode May lose some FOV

8 MCU ELECTRONICS DESIGN

This Section is intended to describe the features and the architecture of the Subsystem dedicated electronics (i.e. electronics circuitry and interconnection including complex circuits like FPGAs and/or dedicated micro-controllers.

8.1 Overall description.




Motor (4 wires)
LVDT (8 wires)
Launch latch (6 wires)
T° probes (8 wires)
Position encoder
2 x 8 wires
MDM type
25 S
25 P
T° probes (8 wires)
Launch latch (6 wires)
LVDT (8 wires)
Motor (4 wires)
Conection board
Preamplifier
Conection board
Prime
25 S
25 P
MDM type
Redundant
37 S
37 P
Mechanism
Prime
Prime
Redundant
Redundant
I/F with RF box = TBD
Laboratoire d'Astrophysique de Ma
Title
SMEC/FTS/mechanism - INTERCONNI
Size
A4
Document Number
LAM/ELE/FTS/GEN/00-12
Date:
Sheet 1 of

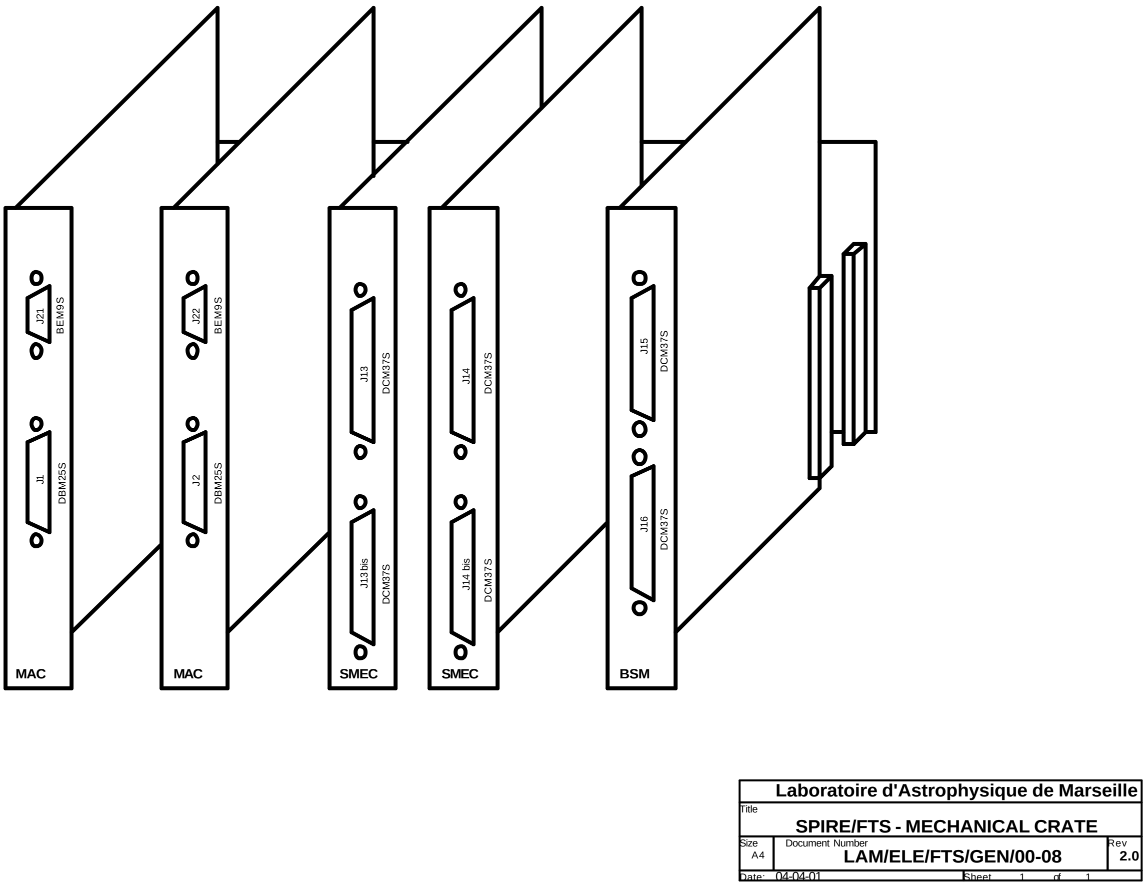
8.2 MCU Mechanical implementation.

8.2.1 Mechanism control unit (MCU)

WE includes :

- 2 Multi Axis Controllers (MAC) boards,
- 2 SMEC boards,
- 1 BSM board.

Two MAC boards, SMEC boards and BSM board are plugged on the same mother board.



Dimensions of the MAC, SMEC and BSM boards are 220 x 223 mm² (TBC).

8.2.2 Position encoder preamplifier box

The preamplifier box include the two preamplifiers for main and redundant channels. It is located in the SMEC mechanism, near the position encoder heads.

Dimensions and mass are TBD

8.2.3 Mass

The mass assessment is globally TBD.

Subsystem	Number	Nominal mass / unit (g)	Margin / unit (g)	Dispersion / unit (g)	Total max. mass (g)	Remarks
MAC module board	2	TBD	TBD	TBD	TBD	
MAC module mechanical and thermal parts	2	TBD	TBD	TBD	TBD	
SMEC module board	2	TBD	TBD	TBD	TBD	
SMEC module mechanical and thermal parts	2	TBD	TBD	TBD	TBD	
BSM module board	2	TBD	TBD	TBD	TBD	
BSM module mechanical and thermal parts	2	TBD	TBD	TBD	TBD	
preamplifier board	2	TBD	TBD	TBD	TBD	
TOTAL					**TBD**	

8.2.4 Power dissipation

Subsystem	Voltage	Average power (TBC)	Peak power / time (TBC)	Average / peak current	Remarks
MAC board DSP and logic	+ 5 V	8000 mW (TBC)	13500 mW (TBC) t = 100 ms (TBC)	1.6 A / 2.7 A (TBC)	• power peak happens during DSP boot
MAC board analogue & data converters	± 13 V	1050 mW	1050 mW	50 mA on + 13 V 30 mA on – 13 V	
SMEC board analogue electronics	± 13 V	2000 mW (TBC)	2000 mW (TBC)	80 mA on + 13 V 80 mA on – 13 V	
SMEC board PWR Amplifier	± 15 V (TBC)	500 mW on + 15V and 500 mW on - 15V	3000 mW	33 mA 100 mA max (TBC)	• motor harness resistance less than 20 ohms
BSM board analogue electronics	± 13 V	1400 mW (TBC)	1400 mW (TBC)	55 mA on + 13 V 50 mA on – 13 V	
BSM board PWR Amplifier	± 15 V (TBC)	450 mW on + 15V or 450 mW on - 15V (TBC)	1500 mW on + 15V or 1500 mW on -15V t < 10 ms	30 mA 100 mA max (TBC)	• motor harness resistance less than 200 ohms • power peak happens during acceleration of the mirror.
TOTAL		13900 mW	22450 mW		• power peaks on SMEC and MAC boards are never at the same time.

8.3 Interfaces

8.3.1 MCU I/F with SMEC

I/F	Type	Nb. of cond.	I/O	Value	Comments
Linear Motor Coil	analogue current	2	out	100 mA max. (TBC)	20 Ω load for total harness resistance.
	back EMF analogue voltage	2	in	10 µA max.	
Optical Encoder	Sinusoidal low level signals provided and preamplified by 3 photodiodes delivering moire fringes signals	12	in	10 µA peak (TBC)	distance from position encoder to preamplifier as short as possible.
	analogue current delivered to LED	2	out	0.5 mA DC (TBC)	3 levels: 0,0.25 and 0.5 mA
	analogue current delivered to preamplifier	2	out	200 µA DC (TBC)	
Temperature probe	analogue current delivered to t° probe	2	out	10 µA DC (TBC)	
	analogue t° signals provided by t° probe	2	in	TBD mV	
LVDT	Analog voltage	2	out	1 mA peak	
	Analog voltage	4	in	10 µA max	

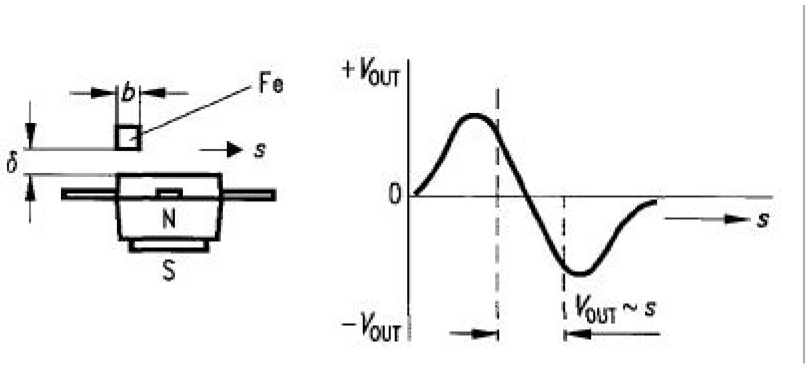
8.3.2 MCU I/F with BSM

I/F	Type	Nb.	I/O	Value	Constraints
Voice Coil Motor	analogue current	2 x 2 motors	out	50 mA max (TBC)	
Home Position magnetoresistive sensors	analogue current delivered to resistors bridge	3 x 2 bridges	out	2 x 1 mA (TBC)	
	analogue voltage delivered by resistors bridge	2 x 2 bridges	in	+/-100 mV	

Position sensor specification :

- resistance at 300K : 2 x 100 Ω nominal
- small change at 4K
- excitation current : 1 mA DC
- dissipation : 0.5 mW (constant)
- output voltage : +-0.1 V
- Hysteresis : ~ 0.05° - corrected as part of control algorithm

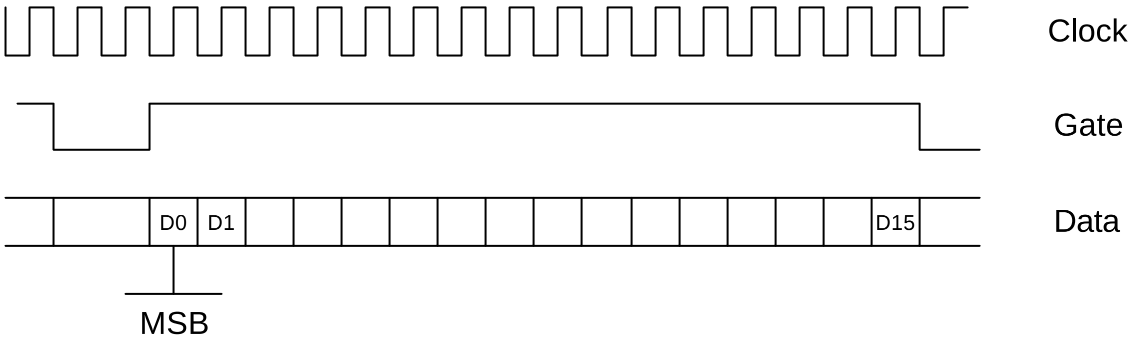
- Infineon (ex-Siemens) FP 212 L100-22 differential field plates sensing moving soft iron pieces
- The sensors are dual InSb/NiSb magnetoresistive elements, biased with a permanent magnet and forming part of a bridge circuit.



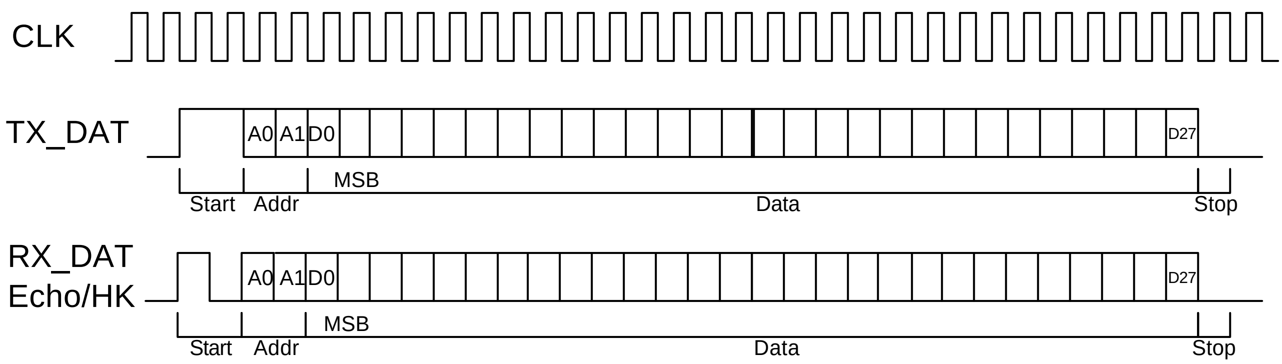
8.3.3 MCU I/F with DPU

The interface with the SPIRE DPU consists of 2 synchronized serial links :

- the command serial link for configuration order and low speed exchanges with the DPU
- the telemetry / HK serial link for high-speed data transfer to the DPU.



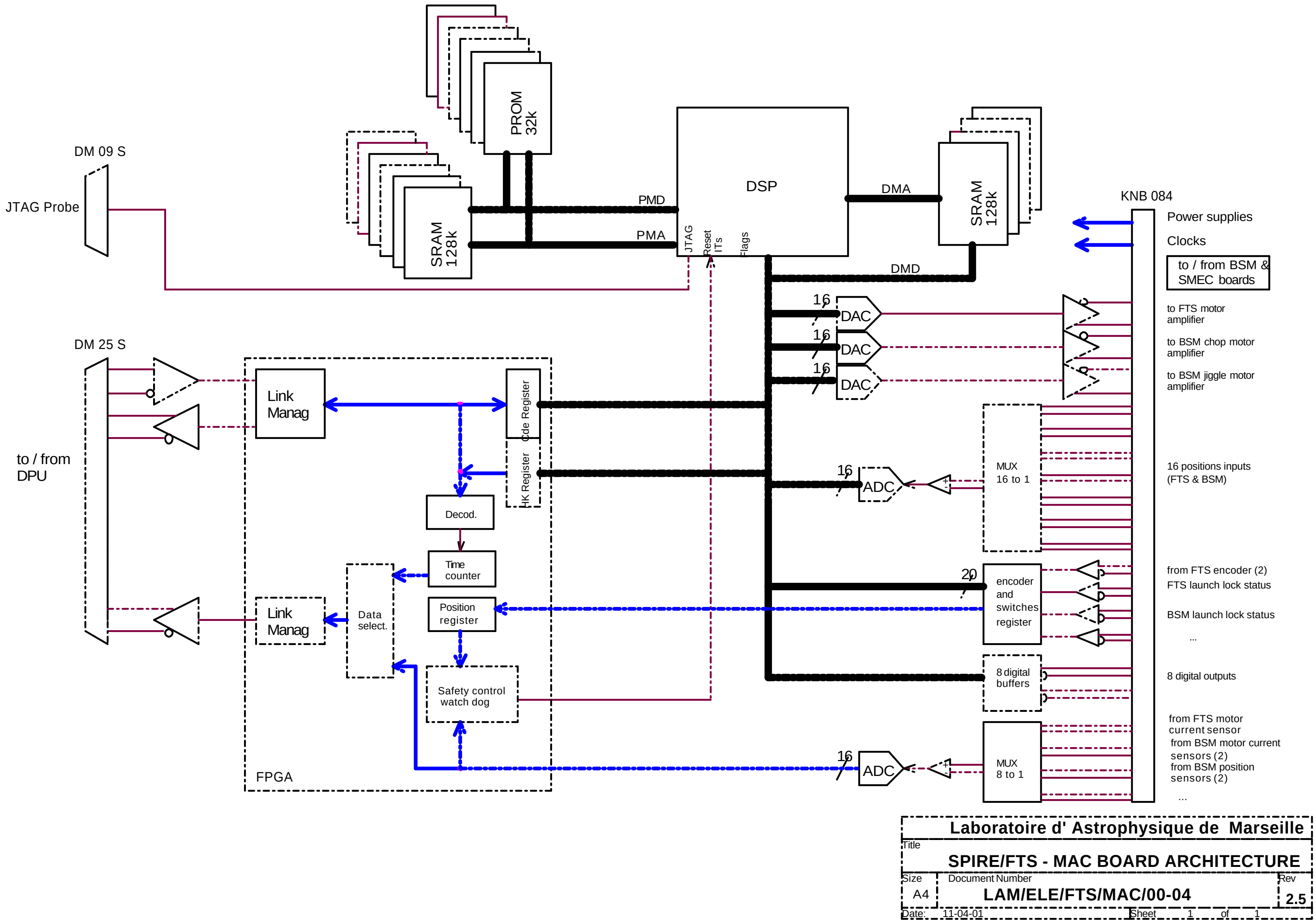
High speed interface protocol



FSDPU/FSDRCU Low Speed Interface protocol

8.4 Board Design description.

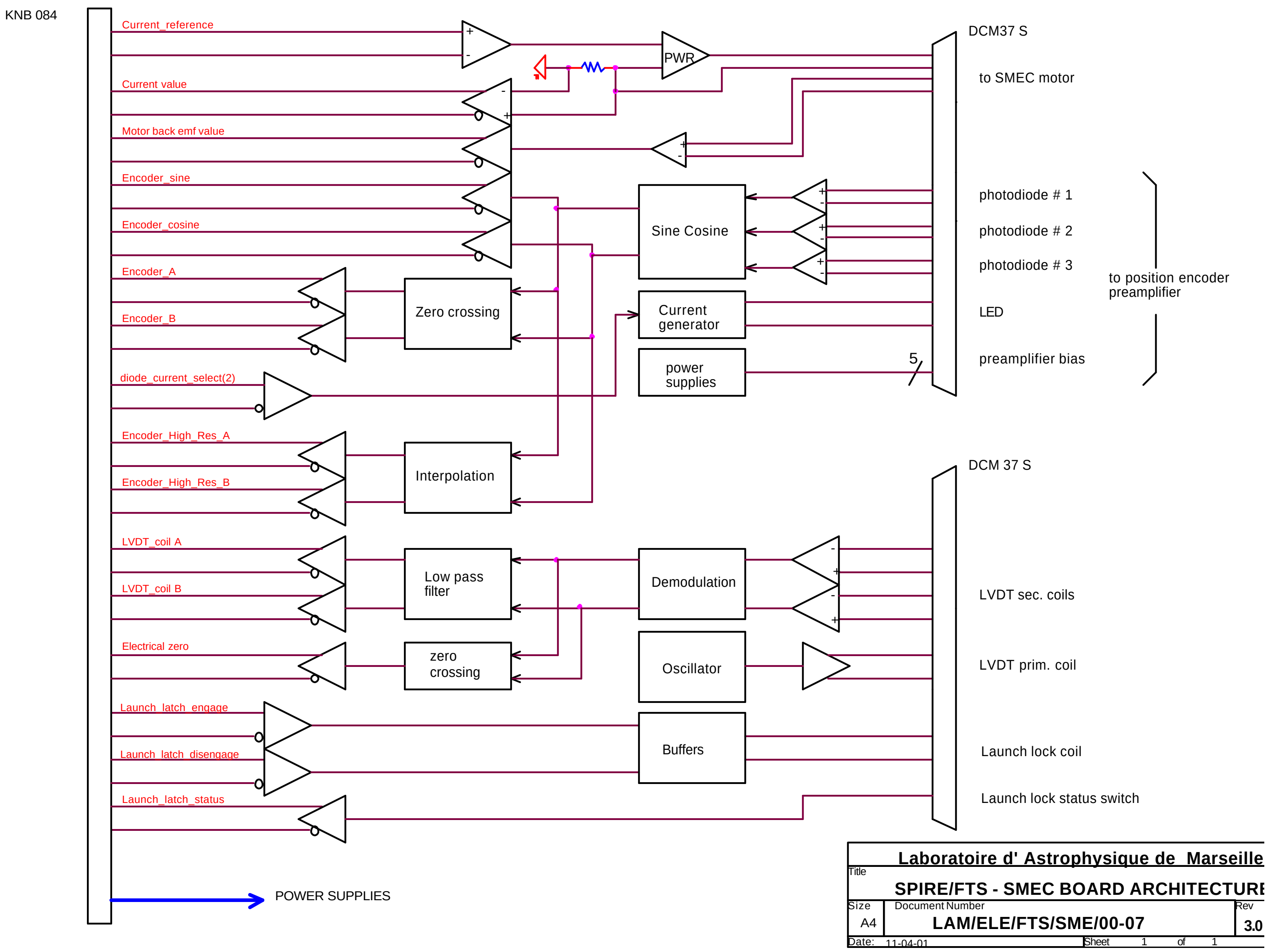
8.4.1 MAC Board



8.4.2 SMEC Board

The SMEC Board is a 220x223 mm2 card including all analog electronics interfacing the SMEC Subsystem. It includes:

- the motor amplifier
- the optical encoder acquisition
- the LVDT conditioner
- the launch lock drivers



8.4.2.1 MOTOR Power amplifier

The Power Amp is based on bipolar transistors (complementary pairs of 2N5153/2N5154) providing a gain of 10 mA/V with a bandwidth of 3.3 kHz.

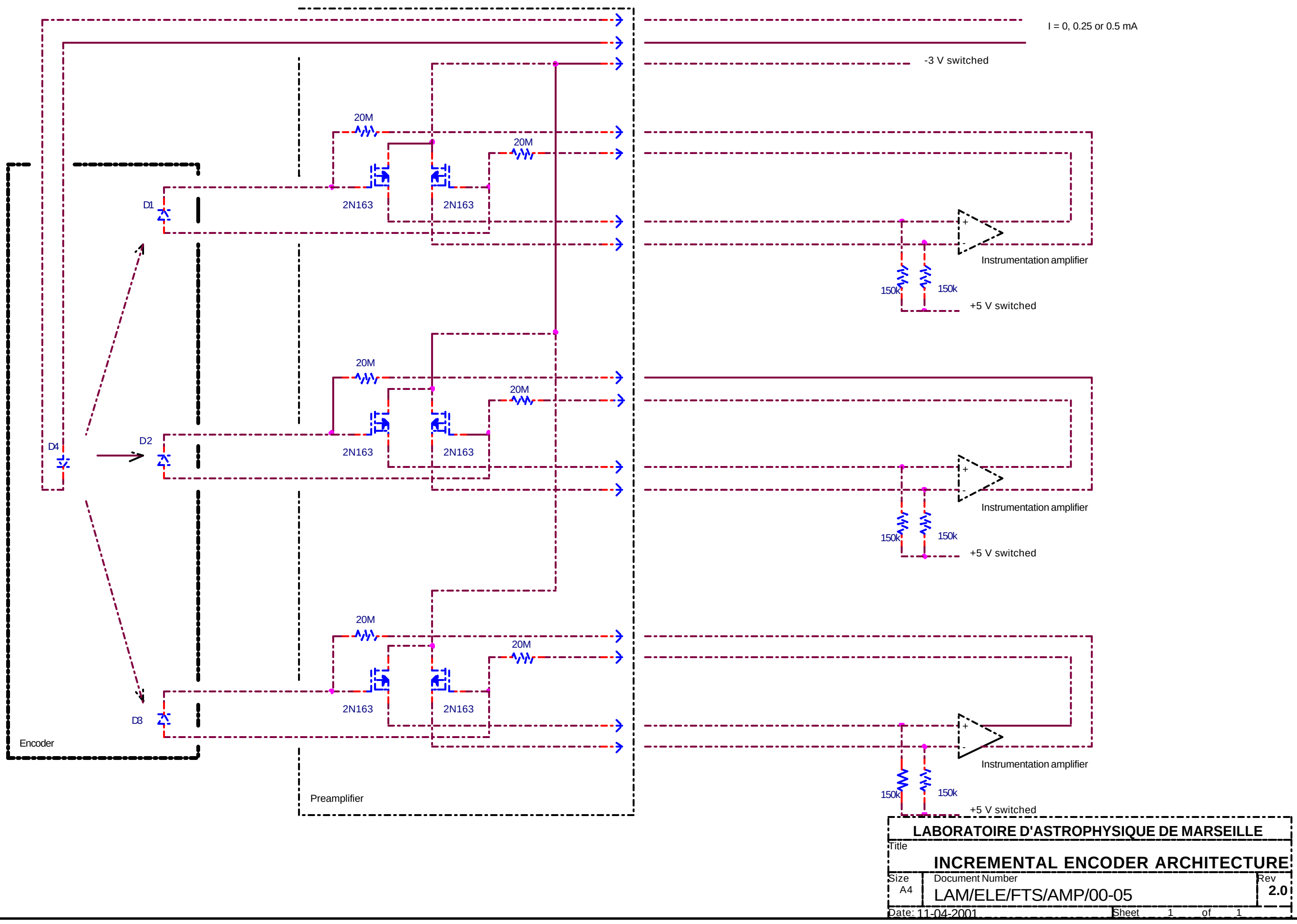
The layout is To Be inserted

8.4.2.2 Encoder acquisition electronics AND PREAMP

see SMEC Board Layout for encoder acquisition electronics.

8.4.2.2.1 Optical Encoder Preamplifier

- The preamplifier board includes the two preamplifiers for main and redundant channels. It is located in the SMEC mechanism, near the position encoder heads.
- 6 x 3N163 PChannel Mosfet + 6 redundant
- Total dissipation for preamp < 500 uW
- Total dissipation for encoder head < 1 mW



8.4.3 BSM Board

Position sensor read-out :

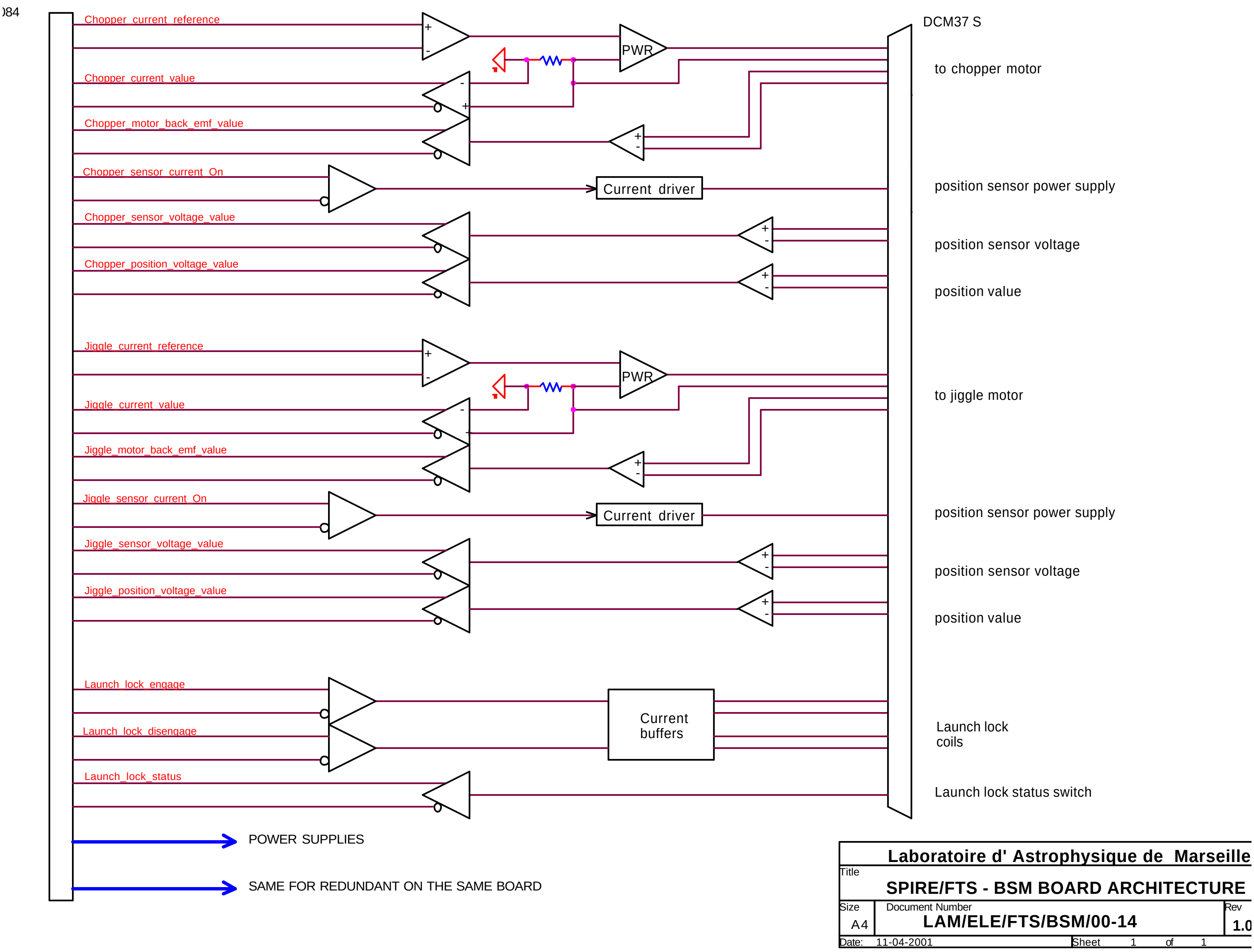
- Bridge circuit with switchable DC bias,
- Instrumentation Amplifier eg A524,
- Multiplexer and 16 bit A/D converter eg 7805ALPRP (SEi)

Motor drive:

- Complimentary bipolar transistor pair eg 2N 5153/4

Motor Current Measurement:

- series resistor and connection to A/D converter



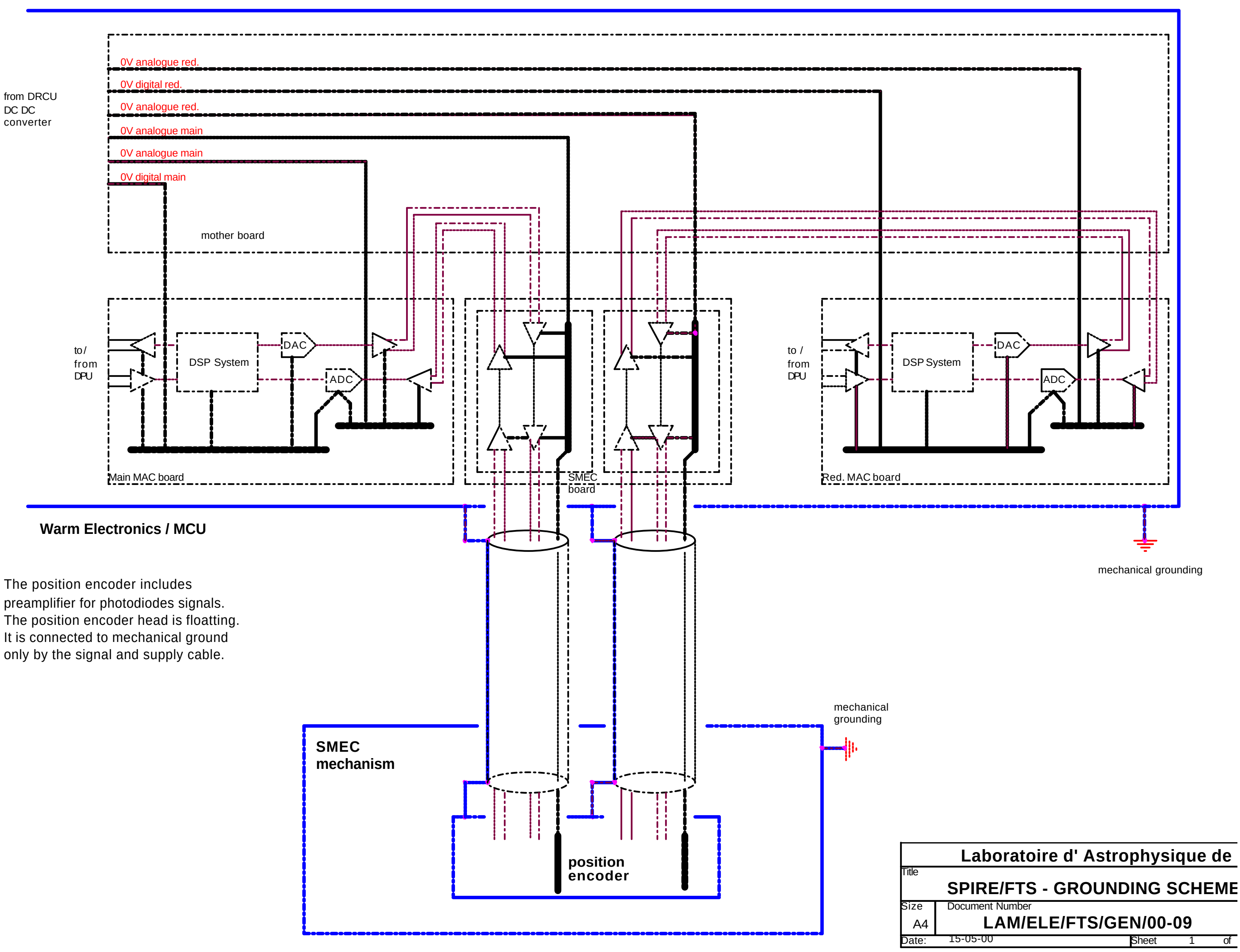
8.5 Power Supply

Subsystem	Nominal voltage	Average current	Peak current / time	Voltage stability	Voltage Noise	Remarks
MAC board DSP and logic	+ 5 V	1.6 A (TBC)	2.7 A (TBC) / 100 ms (TBC)	± 2.5 %	TBD	
MAC board analogue elec. & data converters	± 13 V	50 mA on + 13 V 30 mA on – 13 V	-	± 20 %	TBD	
SMEC board analogue electronics	± 13 V	50 mA on + 13 V 50 mA on – 13 V	80 mA on + 13 V 70 mA on – 13 V	± 20 %	TBD	includes power for preamplifiers and position encoder.
SMEC board PWR Amplifier	+/-15 V (TBC)	3 mA (TBC)	100 mA (TBC)	± 5 %	TBD	motor harness resistance < 20 ohms
BSM board analogue electronics	± 13 V	55 mA on + 13 V 50 mA on – 13 V	-	± 20 %	TBD	includes power for position encoder.
BSM board PWR Amplifier	+/-15 V (TBC)	30 mA (TBC)	100 mA (TBC) / 50 ms (TBC)	± 5 %	TBD	

- Two complete sets of these above power lines are needed (main power lines & redundant power lines).
- Encoder preamplifier boards (M & R) are powered by SMEC board.
- All the electrical components in the SMEC mechanism are powered by the SMEC board.
- All the electrical components in the BSM mechanism are powered by the BSM board.

8.6 Grounding Scheme.

- Motor PWR Analog ground to be added on the schematics
- The analog grounds not connected @ MCU level
- Analog/Digital grounds connected @ ADC level
- Analog ground not connected to mechanics

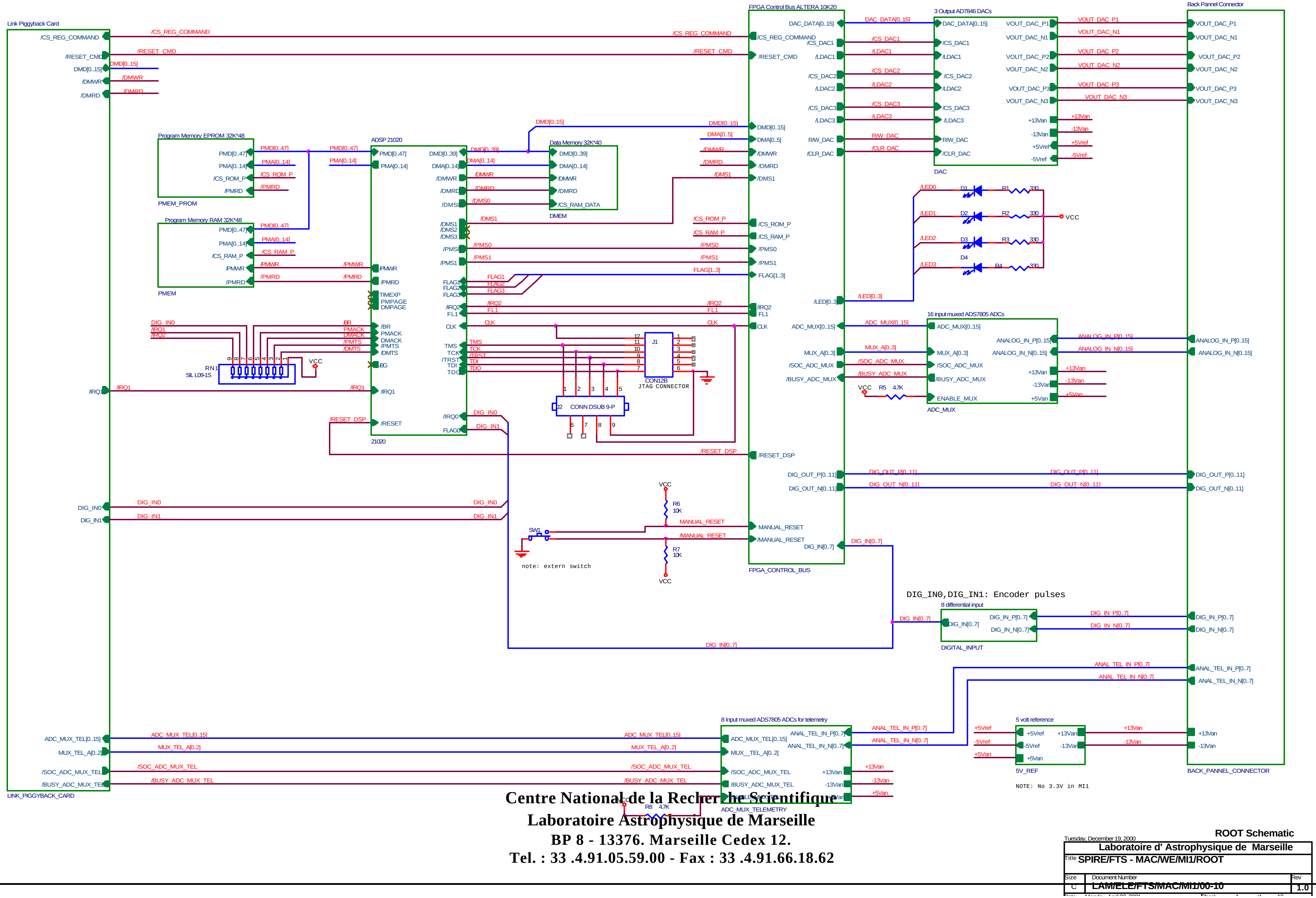


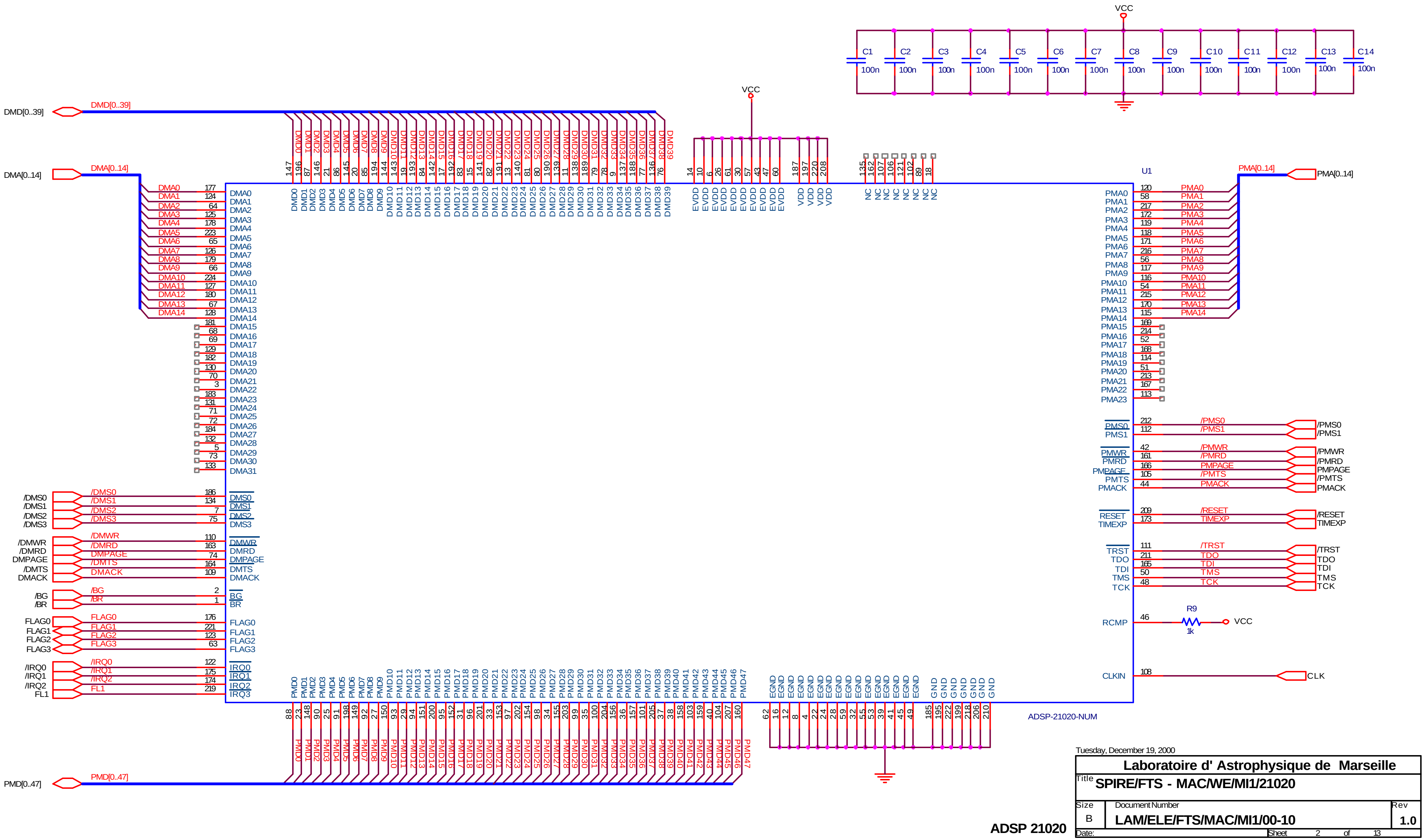
8.7 Use of Special Components/Component list.

See the EEE part list for the complete set of components (doc ref LAM/ELE/FTS/QUA/000201).

8.8 MAC Board Layouts.

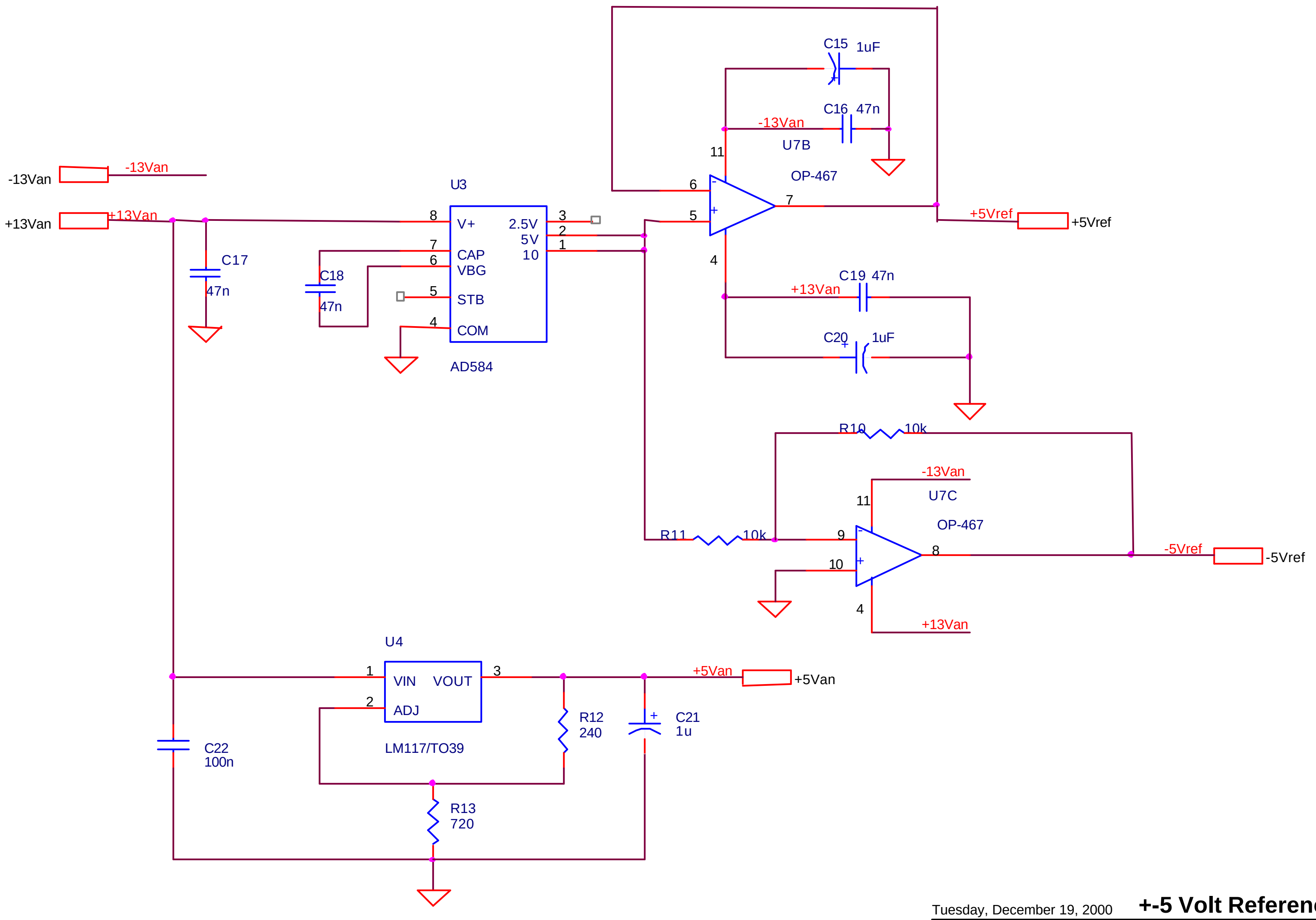
(see next pages).


Link Piggyback Card
LINK_PIGGYBACK_CARD
Program Memory EPROM 32K*48
PMEM_PROM
Program Memory RAM 32K*48
PMEM
ADSP 21020
21020
Data Memory 32K*40
DMEM
FPGA Control Bus ALTERA 10K20
FPGA_CONTROL_BUS
3 Output AD7846 DACs
DAC
16 input muxed ADS7805 ADCs
ADC_MUX
Back Pannel Connector
BACK_PANNEL_CONNECTOR
CON12B
JTAG CONNECTOR
CONN DSUB 9-P
note: extern switch
DIG_IN0,DIG_IN1: Encoder pulses
8 differential input
DIGITAL_INPUT
8 Input muxed ADS7805 ADCs for telemetry
ADC_MUX_TELEMETRY
5 volt reference
5V_REF
NOTE: No 3.3V in MI1
ROOT Schematic
Tuesday, December 19, 2000
Laboratoire d' Astrophysique de Marseille
Title SPIRE/FTS - MAC/WE/MI1/ROOT
Size C
Document Number LAM/ELE/FTS/MAC/MI1/00-10
Rev 1.0



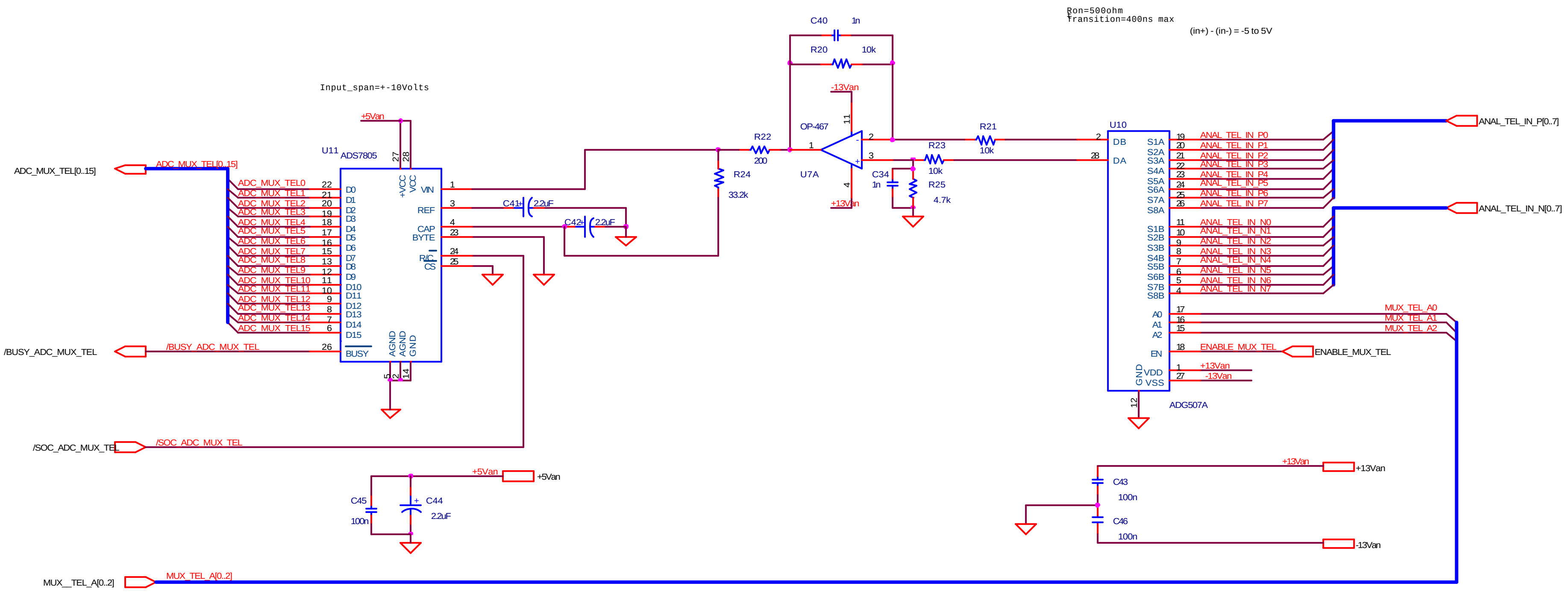
VCC
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14
100n
DMD[0..39]
DMA[0..14]
PMA[0..14]
PMD[0..47]
U1
ADSP-21020-NUM
/DMS0
/DMS1
/DMS2
/DMS3
/DMWR
/DMRD
DMPAGE
/DMTS
DMACK
/BG
/BR
FLAG0
FLAG1
FLAG2
FLAG3
/IRQ0
/IRQ1
/IRQ2
FL1
/PMS0
/PMS1
/PMWR
/PMRD
PMPAGE
/PMTS
PMACK
/RESET
TIMEXP
/TRST
TDO
TDI
TMS
TCK
R9
1k
RCMP
CLKIN
CLK
Tuesday, December 19, 2000
Laboratoire d' Astrophysique de Marseille
Title SPIRE/FTS - MAC/WE/MI1/21020
Size B
Document Number LAM/ELE/FTS/MAC/MI1/00-10
Rev 1.0
Date:
Sheet 2 of 13

ADSP 21020



Tuesday, December 19, 2000 **+-5 Volt References**

Laboratoire d' Astrophysique de Marseille		
Title **SPIRE/FTS - MAC/WE/MI1/5V_REF**		
Size A	Document Number LAM/ELE/FTS/MAC/MI1/00-10	Rev 1.0




```
NOTE 1
According to Burr Brown Data sheet:
1) analog and digital power pins tied to the same +5V power supply, that can be produced from an analog supply, not the digital supply
2) All the grounds tied to the analog ground plane, separated from the system's digital logic ground

NOTE 2:
No hardware trim for calibration: offset and gain corrected in software (TBD?)

NOTE 3:
To read a value from ADC follow the steps:
1) write value on  MUX_TEL_A[0..2] to select channel
2) wait 400ns min, Transition time of ADG507
3) launch a start of conversion (/SOC_ADC_MUX_TEL=0)
4) read /BUSY_ADC_MUX until /BUSY_ADC_MUX_TEL=1
5) read ADC_MUX_TEL[0..15]
the steps 1),2) and 3) are performed by ADSP21020
the steps 4) and 5) are performed by FPGA Control bus,
 so the /BUSY_ADC_MUX_TEL=0 to 1 will be the write
signal
 into a internal register of FPGA Control Bus
```

8 input muxed ADS7805 ADCs for telemetry

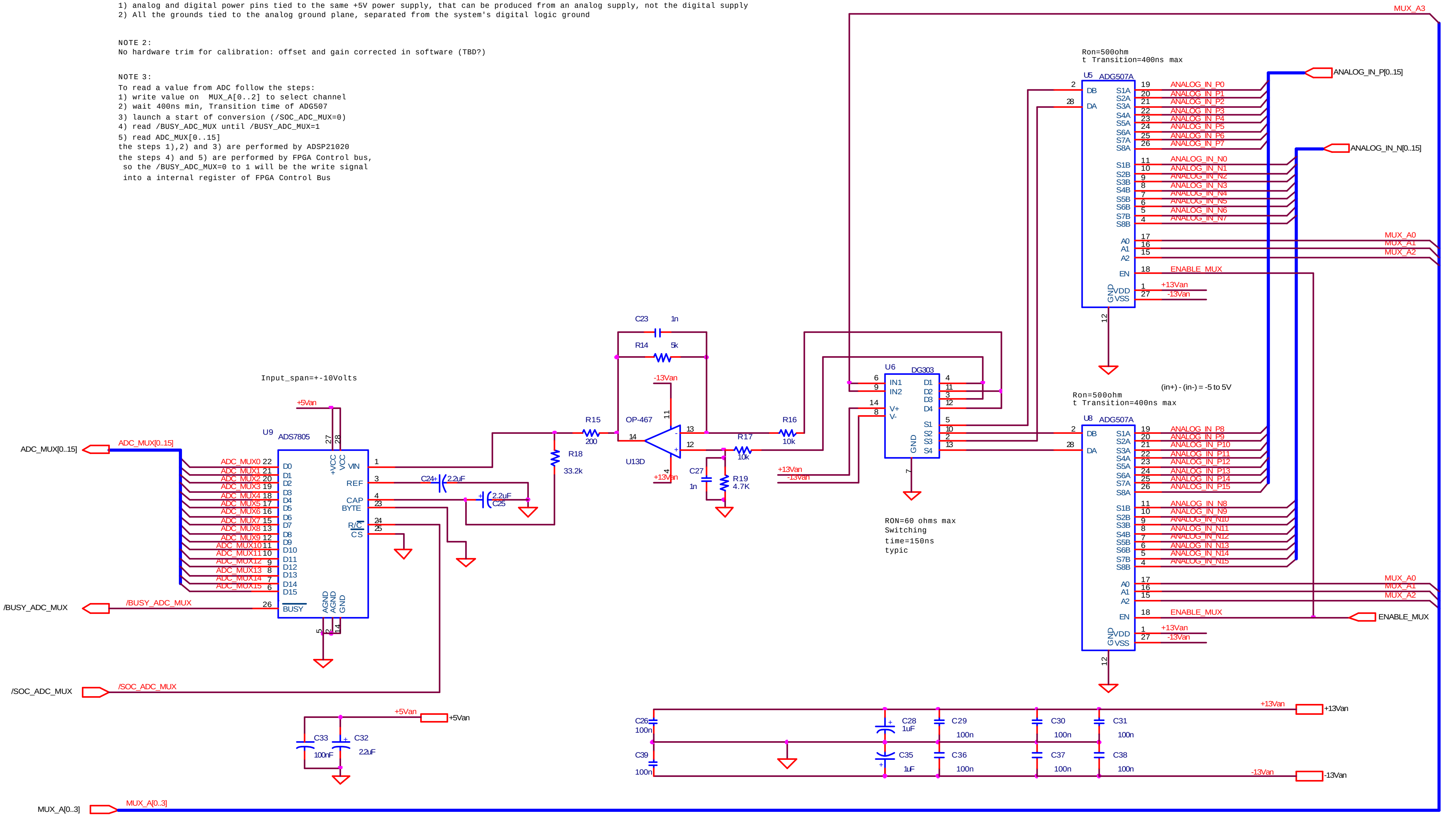
Tuesday, December 19, 2000

Laboratoire d' Astrophysique de Marseille

NOTE 1
According to Burr Brown Data sheet:
1) analog and digital power pins tied to the same +5V power supply, that can be produced from an analog supply, not the digital supply
2) All the grounds tied to the analog ground plane, separated from the system's digital logic ground

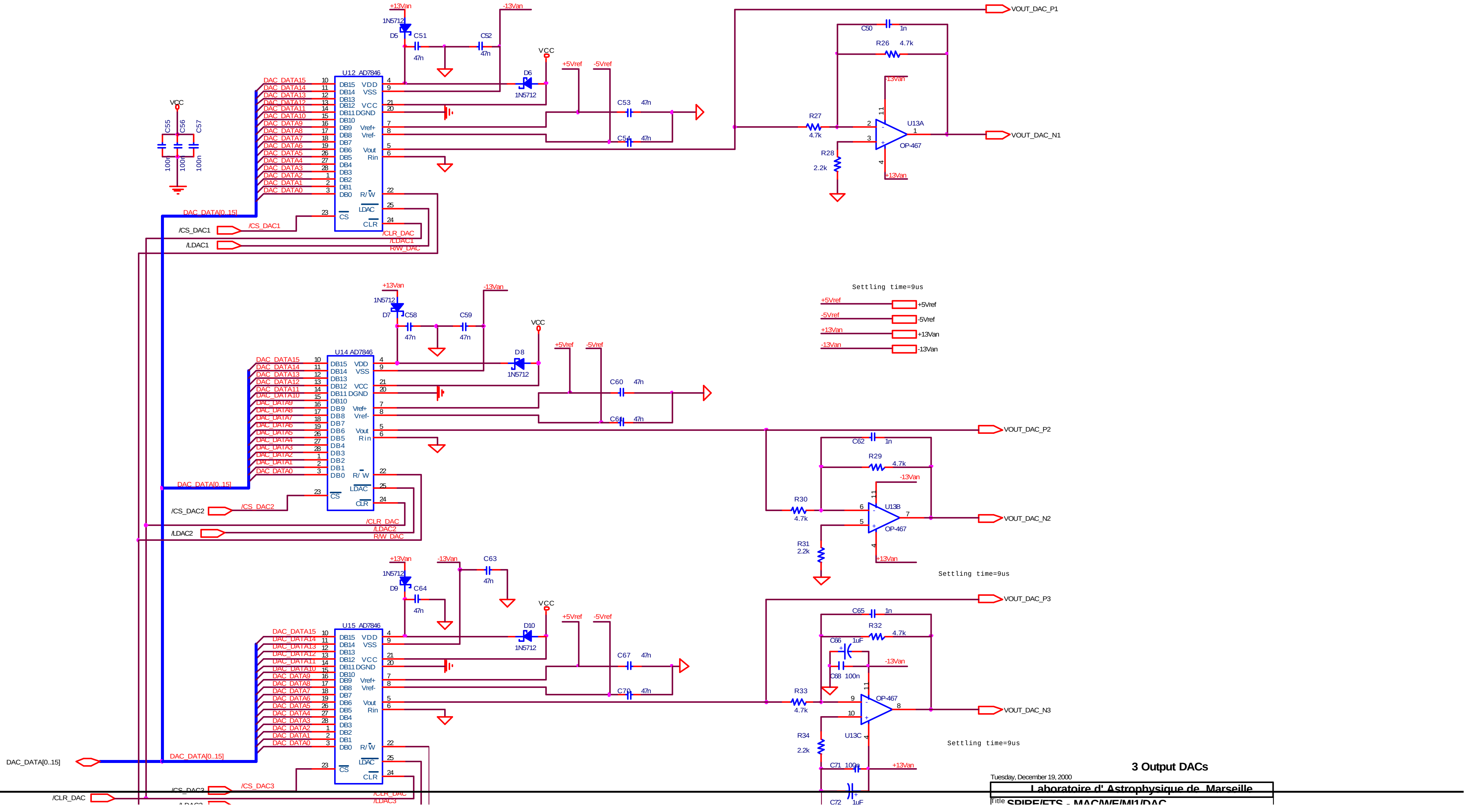
NOTE 2:
No hardware trim for calibration: offset and gain corrected in software (TBD?)

NOTE 3:
To read a value from ADC follow the steps:
1) write value on MUX_A[0..2] to select channel
2) wait 400ns min, Transition time of ADG507
3) launch a start of conversion (/SOC_ADC_MUX=0)
4) read /BUSY_ADC_MUX until /BUSY_ADC_MUX=1
5) read ADC_MUX[0..15]
the steps 1),2) and 3) are performed by ADSP21020
the steps 4) and 5) are performed by FPGA Control bus,
so the /BUSY_ADC_MUX=0 to 1 will be the write signal
into a internal register of FPGA Control Bus

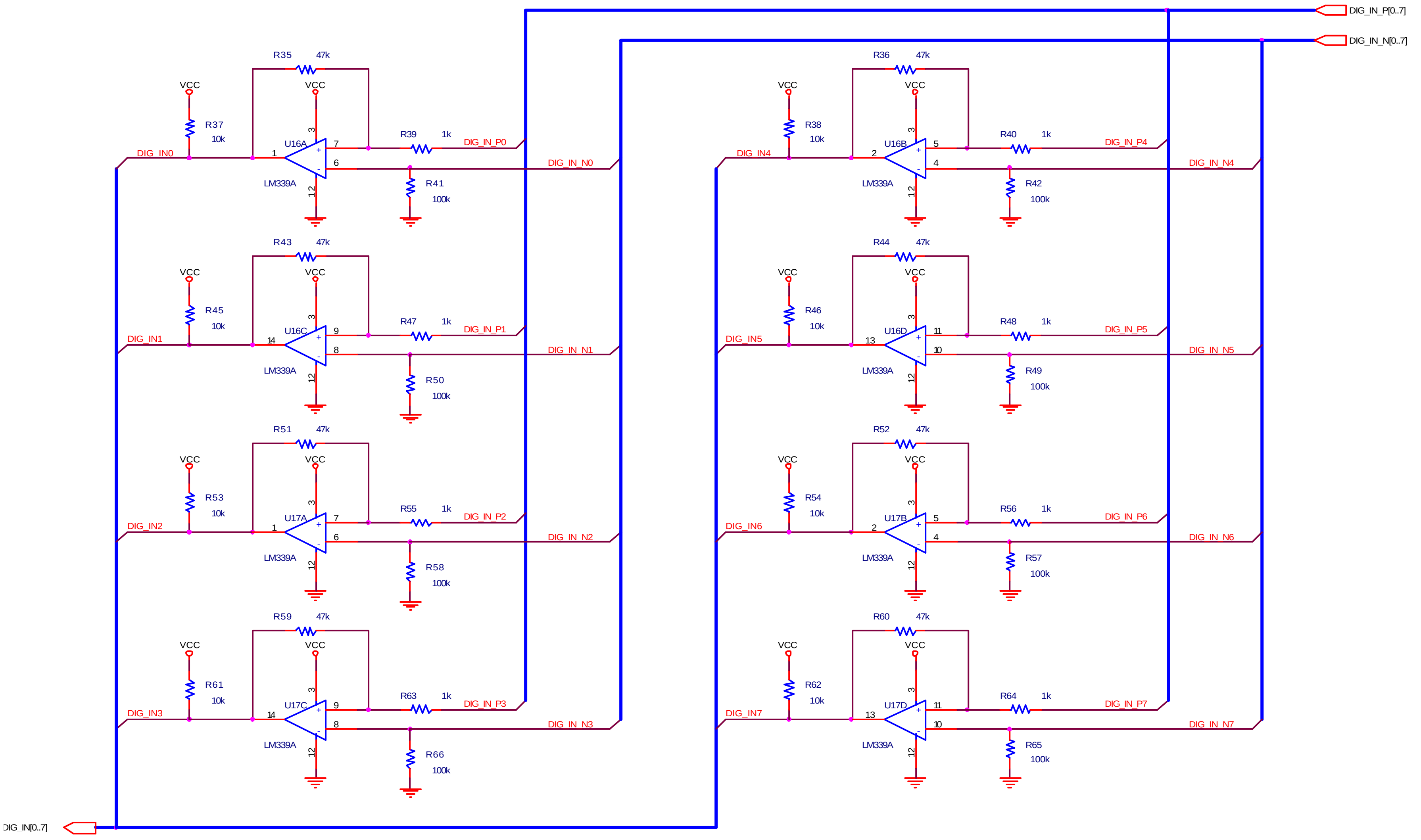


Tuesday, December 19, 2000 — 8 input ADCs MUX

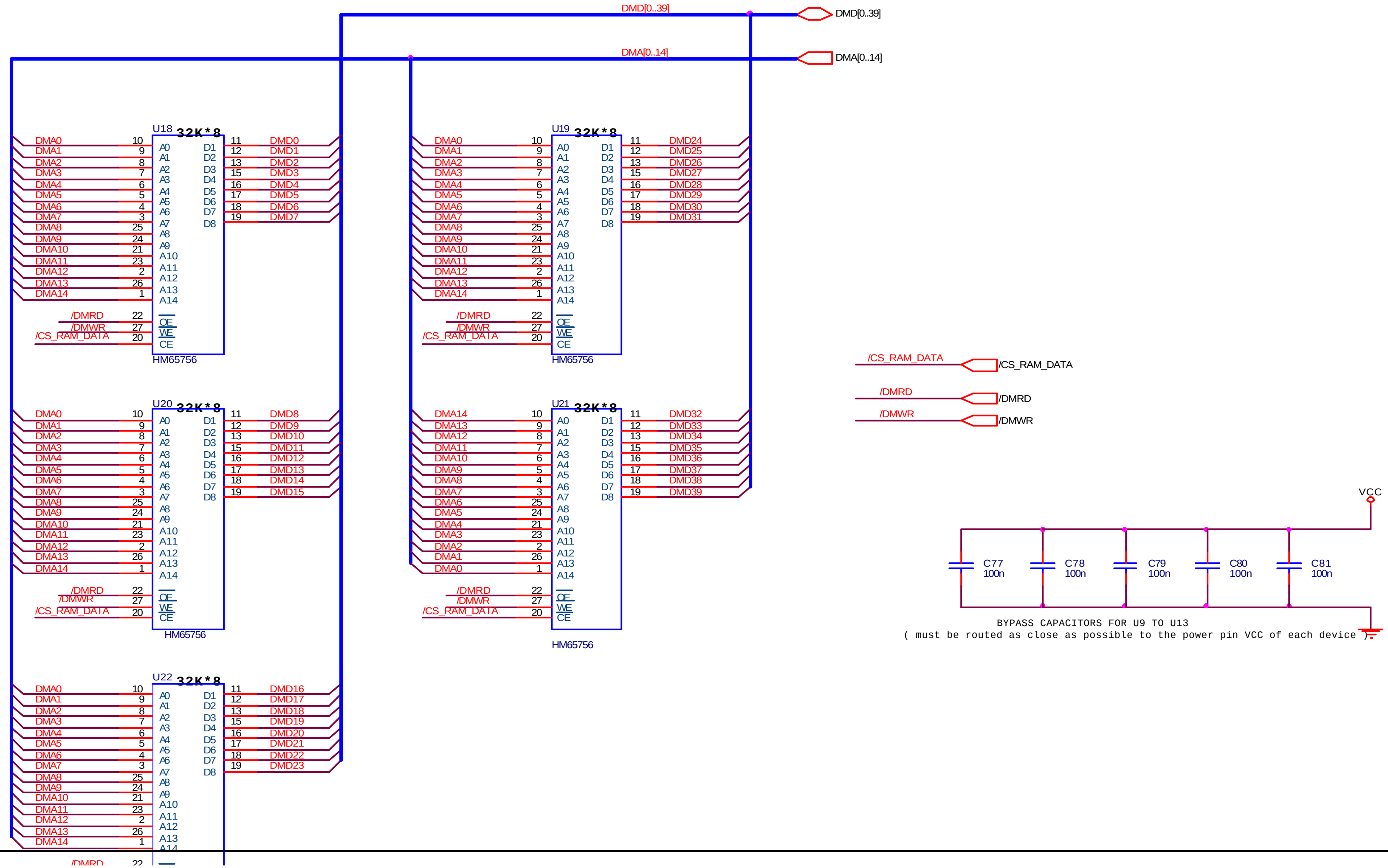
Laboratoire d' Astrophysique de Marseille	
Title: SPIRE/FTS - MAC/WE/MI1/ADC_MUX	
Size A3 — Document Number: LAM/ELE/FTS/MAC/MI1/00-10	Rev 1.0
Date:	Sheet 4 of 13

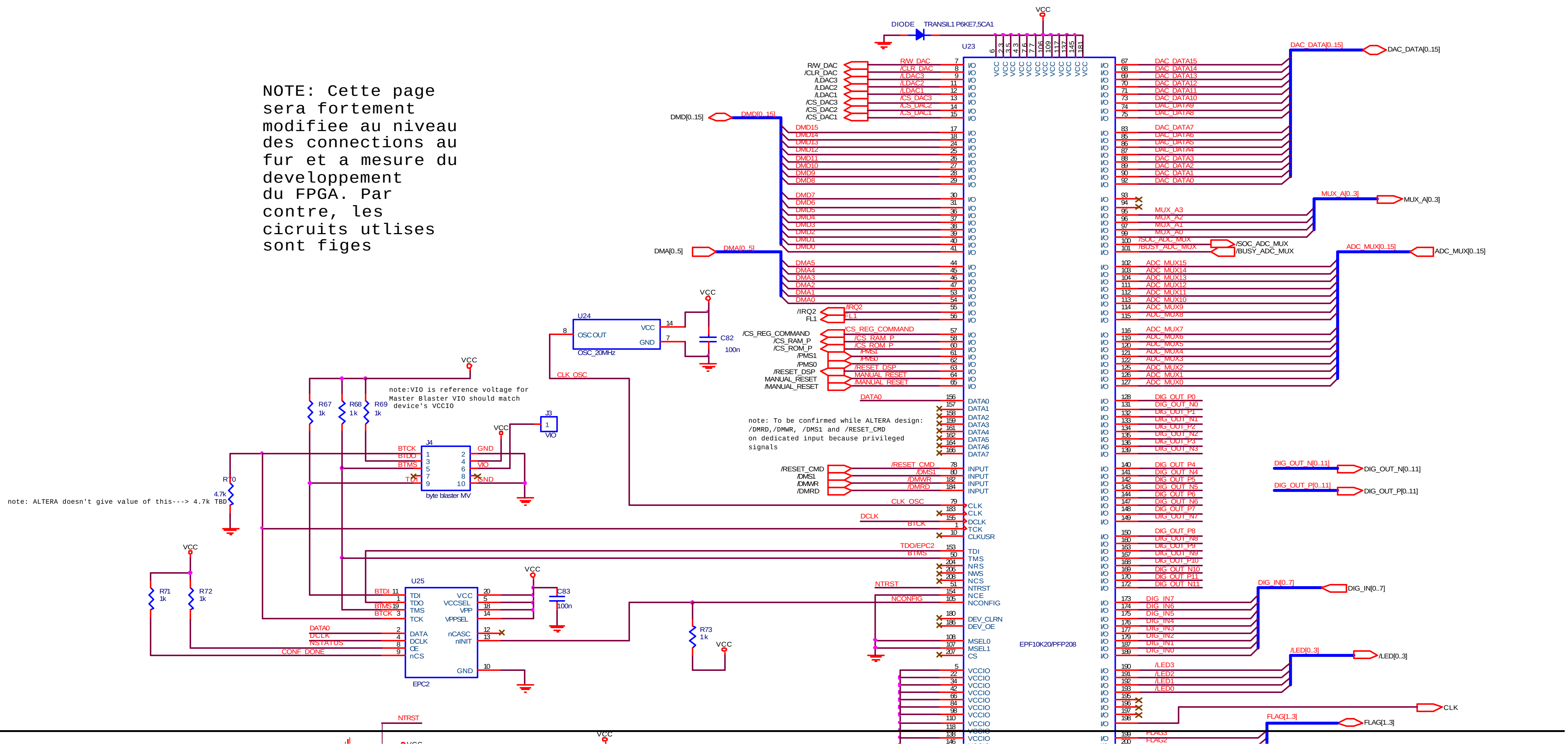


U12 AD7846
U14 AD7846
U15 AD7846
U13A
U13B
U13C
OP-467
VOUT_DAC_P1
VOUT_DAC_N1
VOUT_DAC_P2
VOUT_DAC_N2
VOUT_DAC_P3
VOUT_DAC_N3
DAC_DATA[0..15]
/CS_DAC1
/LDAC1
/CS_DAC2
/LDAC2
/CS_DAC3
/CLR_DAC
R/W_DAC
Settling time=9us
+5Vref
-5Vref
+13Van
-13Van
VCC
1N5712
3 Output DACs
Tuesday, December 19, 2000
Laboratoire d' Astrophysique de Marseille
Title SPIRE/FTS - MAC/WE/MI1/DAC

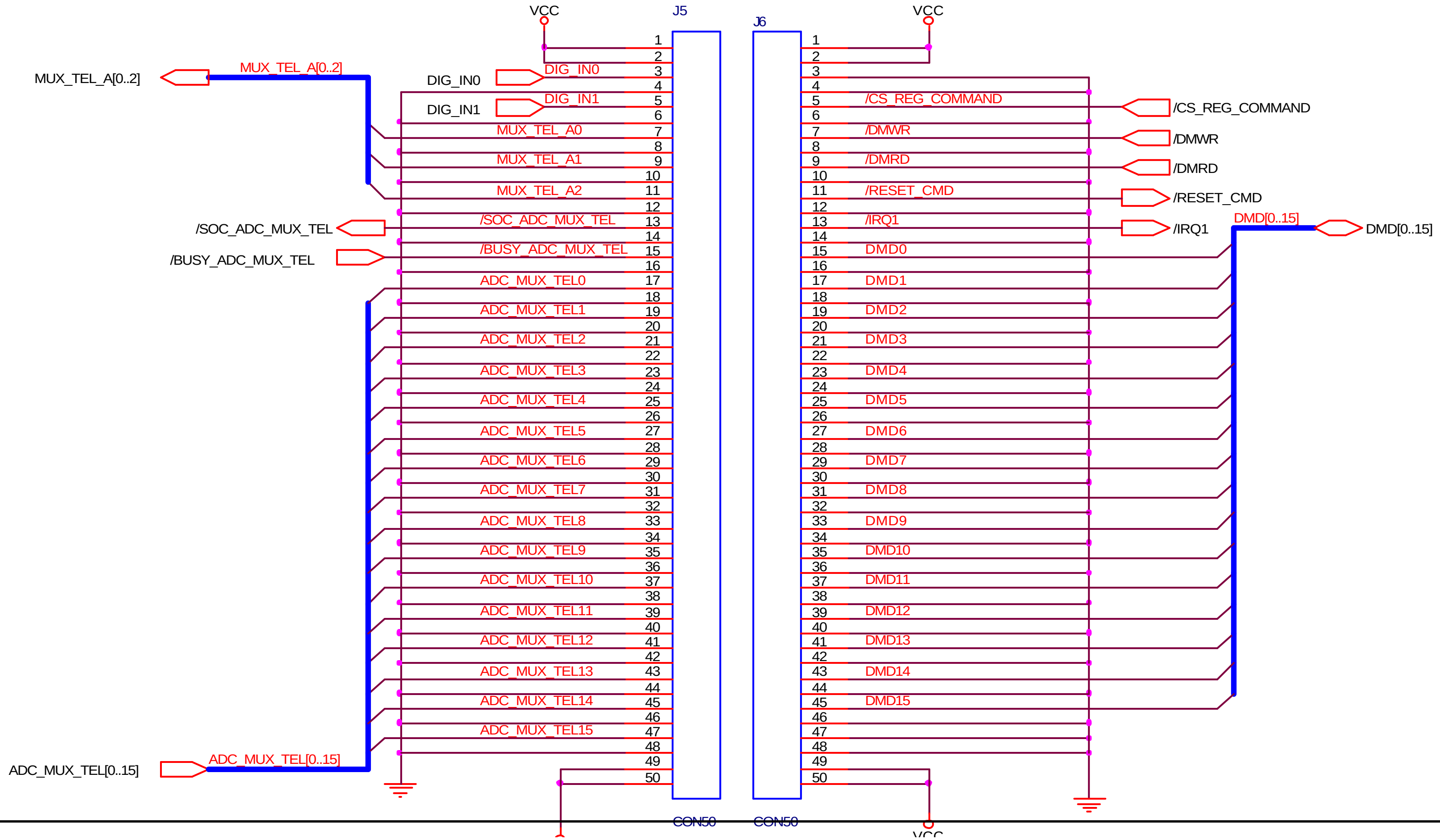


DIG_IN_P[0..7]
DIG_IN_N[0..7]
DIG_IN[0..7]
R35 47k
R36 47k
R37 10k
R38 10k
R39 1k
R40 1k
R41 100k
R42 100k
R43 47k
R44 47k
R45 10k
R46 10k
R47 1k
R48 1k
R49 100k
R50 100k
R51 47k
R52 47k
R53 10k
R54 10k
R55 1k
R56 1k
R57 100k
R58 100k
R59 47k
R60 47k
R61 10k
R62 10k
R63 1k
R64 1k
R65 100k
R66 100k
U16A
U16B
U16C
U16D
U17A
U17B
U17C
U17D
LM339A
VCC
DIG_IN0
DIG_IN1
DIG_IN2
DIG_IN3
DIG_IN4
DIG_IN5
DIG_IN6
DIG_IN7
DIG_IN_P0
DIG_IN_P1
DIG_IN_P2
DIG_IN_P3
DIG_IN_P4
DIG_IN_P5
DIG_IN_P6
DIG_IN_P7
DIG_IN_N0
DIG_IN_N1
DIG_IN_N2
DIG_IN_N3
DIG_IN_N4
DIG_IN_N5
DIG_IN_N6
DIG_IN_N7

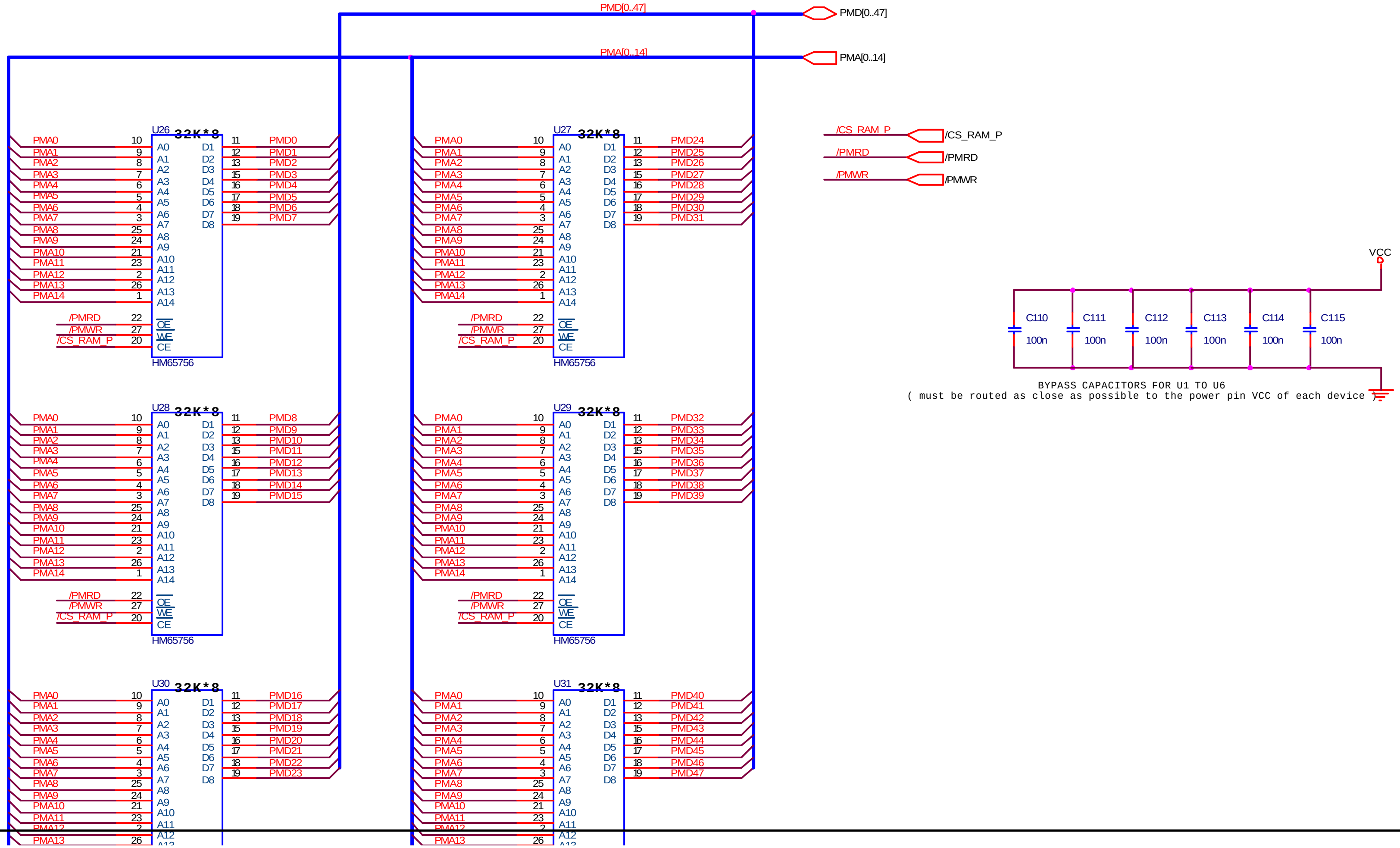

DMD[0..39]
DMA[0..14]
U18 32K*8 HM65756
U19 32K*8 HM65756
U20 32K*8 HM65756
U21 32K*8 HM65756
U22 32K*8
/CS_RAM_DATA
/DMRD
/DMWR
VCC
C77 100n
C78 100n
C79 100n
C80 100n
C81 100n
BYPASS CAPACITORS FOR U9 TO U13
(must be routed as close as possible to the power pin VCC of each device)



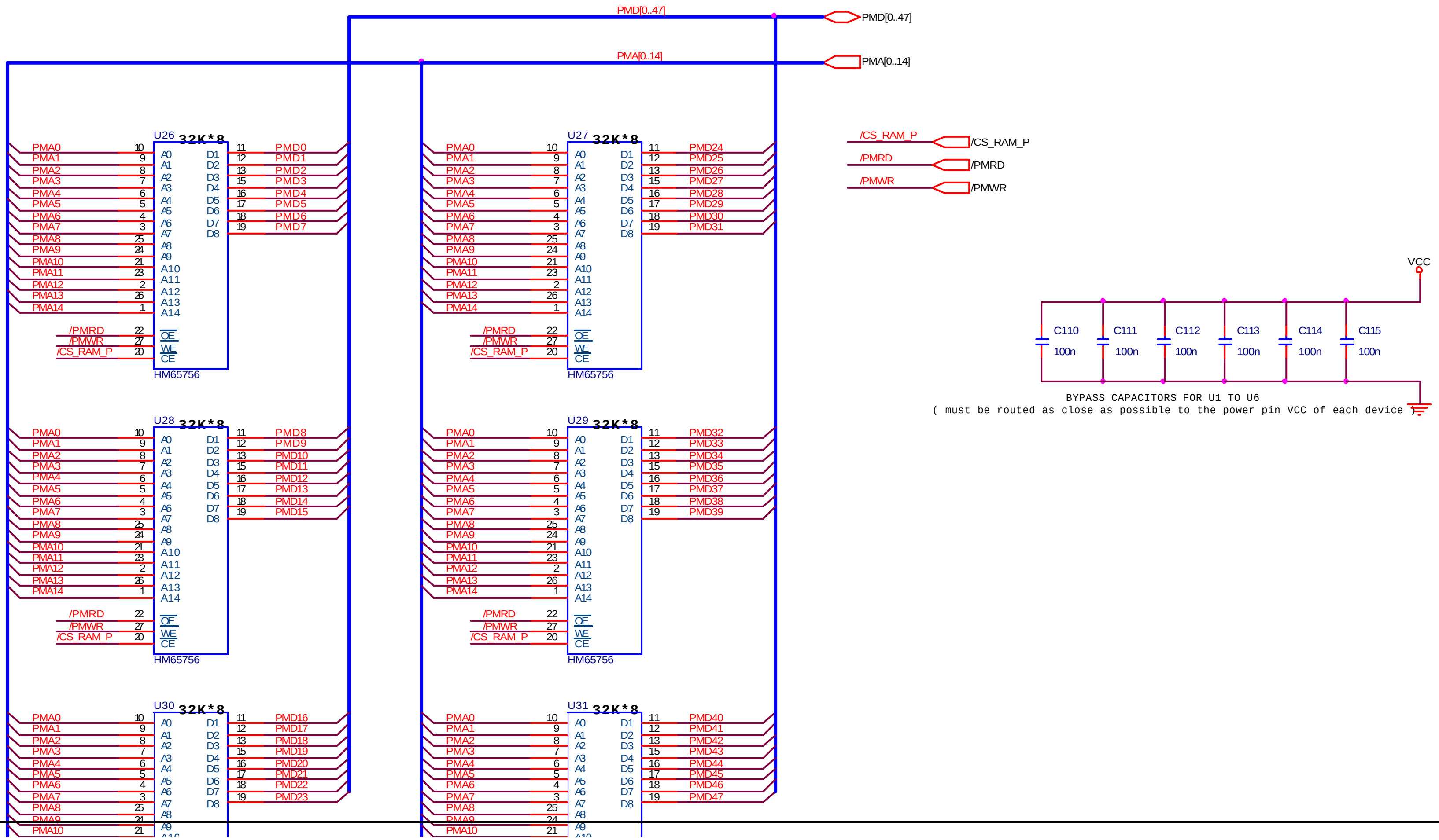
NOTE: Cette page sera fortement modifiee au niveau des connections au fur et a mesure du developpement du FPGA. Par contre, les cicruits utlises sont figes
note:VIO is reference voltage for Master Blaster VIO should match device's VCCIO
note: To be confirmed while ALTERA design: /DMRD,/DMWR, /DMS1 and /RESET_CMD on dedicated input because privileged signals
note: ALTERA doesn't give value of this---> 4.7k TBD
U23
EPF10K20/PFP208
U24
OSC_20MHz
U25
EPC2
J4
byte blaster MV
J3
VIO
DIODE
TRANSIL1 P6KE7,5CA1
DAC_DATA[0..15]
DMD[0..15]
DMA[0..5]
MUX_A[0..3]
ADC_MUX[0..15]
DIG_OUT_N[0..11]
DIG_OUT_P[0..11]
DIG_IN[0..7]
/LED[0..3]
FLAG[1..3]
CLK



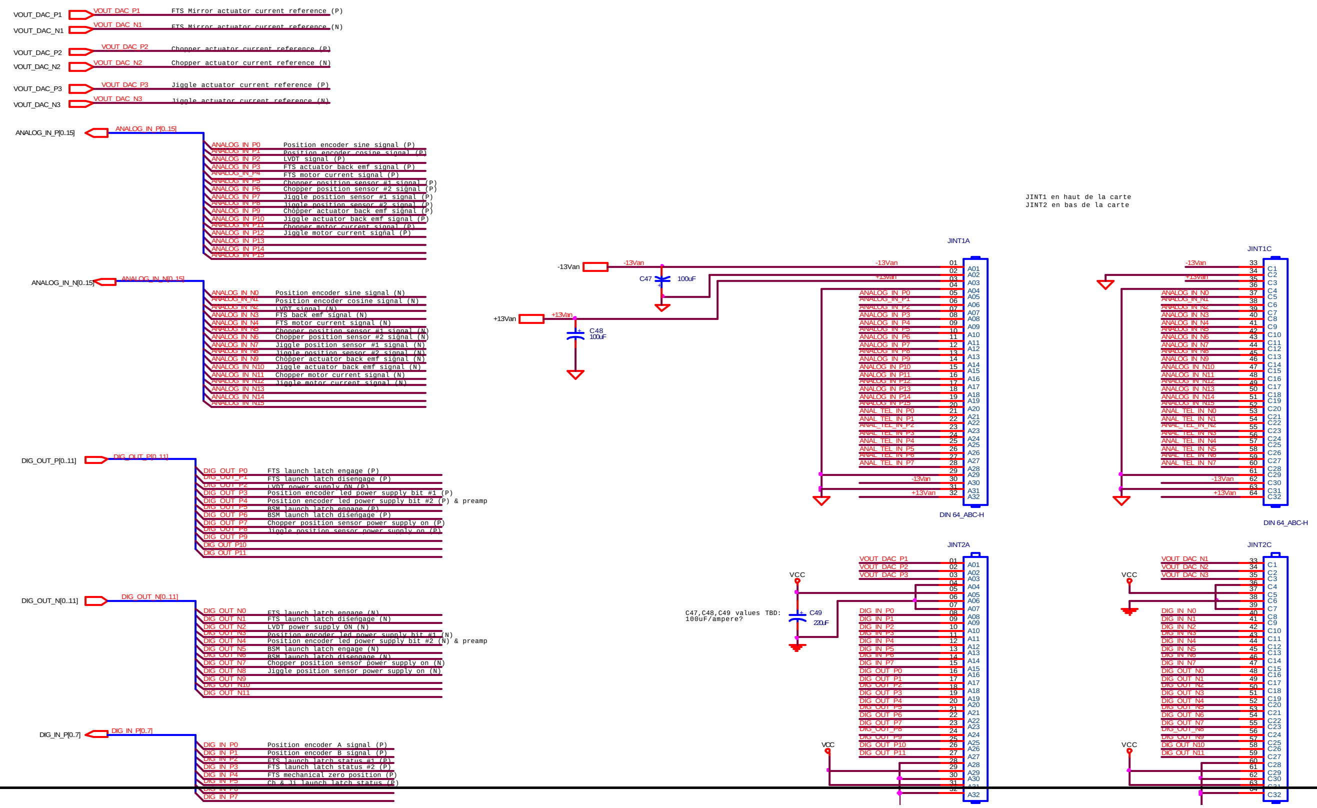
VCC
J5
J6
VCC
MUX_TEL_A[0..2]
MUX_TEL_A[0..2]
DIG_IN0
DIG_IN0
DIG_IN1
DIG_IN1
MUX_TEL_A0
MUX_TEL_A1
MUX_TEL_A2
/SOC_ADC_MUX_TEL
/SOC_ADC_MUX_TEL
/BUSY_ADC_MUX_TEL
/BUSY_ADC_MUX_TEL
ADC_MUX_TEL0
ADC_MUX_TEL1
ADC_MUX_TEL2
ADC_MUX_TEL3
ADC_MUX_TEL4
ADC_MUX_TEL5
ADC_MUX_TEL6
ADC_MUX_TEL7
ADC_MUX_TEL8
ADC_MUX_TEL9
ADC_MUX_TEL10
ADC_MUX_TEL11
ADC_MUX_TEL12
ADC_MUX_TEL13
ADC_MUX_TEL14
ADC_MUX_TEL15
ADC_MUX_TEL[0..15]
ADC_MUX_TEL[0..15]
/CS_REG_COMMAND
/CS_REG_COMMAND
/DMWR
/DMWR
/DMRD
/DMRD
/RESET_CMD
/RESET_CMD
/IRQ1
/IRQ1
DMD[0..15]
DMD[0..15]
DMD0
DMD1
DMD2
DMD3
DMD4
DMD5
DMD6
DMD7
DMD8
DMD9
DMD10
DMD11
DMD12
DMD13
DMD14
DMD15
CON50
CON50



PMD[0..47]
PMA[0..14]
/CS_RAM_P
/PMRD
/PMWR
U26 32K*8 HM65756
U27 32K*8 HM65756
U28 32K*8 HM65756
U29 32K*8 HM65756
U30 32K*8
U31 32K*8
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
D1 D2 D3 D4 D5 D6 D7 D8
OE WE CE
PMD0 PMD1 PMD2 PMD3 PMD4 PMD5 PMD6 PMD7
PMD8 PMD9 PMD10 PMD11 PMD12 PMD13 PMD14 PMD15
PMD16 PMD17 PMD18 PMD19 PMD20 PMD21 PMD22 PMD23
PMD24 PMD25 PMD26 PMD27 PMD28 PMD29 PMD30 PMD31
PMD32 PMD33 PMD34 PMD35 PMD36 PMD37 PMD38 PMD39
PMD40 PMD41 PMD42 PMD43 PMD44 PMD45 PMD46 PMD47
VCC
C110 100n
C111 100n
C112 100n
C113 100n
C114 100n
C115 100n
BYPASS CAPACITORS FOR U1 TO U6
(must be routed as close as possible to the power pin VCC of each device



PMD[0..47]
PMA[0..14]
/CS_RAM_P
/PMRD
/PMWR
U26 32K*8 HM65756
U27 32K*8 HM65756
U28 32K*8 HM65756
U29 32K*8 HM65756
U30 32K*8
U31 32K*8
VCC
C110 100n
C111 100n
C112 100n
C113 100n
C114 100n
C115 100n
BYPASS CAPACITORS FOR U1 TO U6
(must be routed as close as possible to the power pin VCC of each device)



VOUT_DAC_P1
VOUT_DAC_N1
VOUT_DAC_P2
VOUT_DAC_N2
VOUT_DAC_P3
VOUT_DAC_N3
ANALOG_IN_P[0..15]
ANALOG_IN_N[0..15]
DIG_OUT_P[0..11]
DIG_OUT_N[0..11]
DIG_IN_P[0..7]
JINT1 en haut de la carte
JINT2 en bas de la carte
JINT1A
JINT1C
JINT2A
JINT2C
DIN 64_ABC-H
-13Van
+13Van
C47
C48
C49
100uF
VCC
C47,C48,C49 values TBD:
100uF/ampere?