FPGA based digitizer for Mu2e, Final Review, Part B

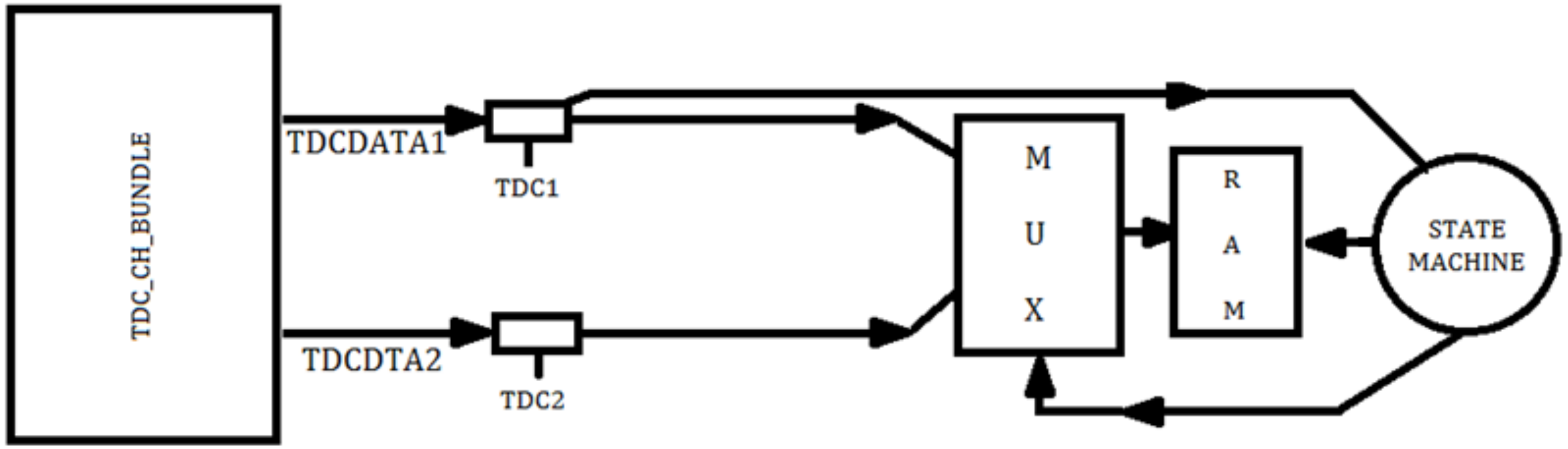
Name : Gabriele Meoni

University: Electrical Engineering at Università di Pisa

Experiment: Mu2e

Field of Training: PPD

Mid Review - Summary

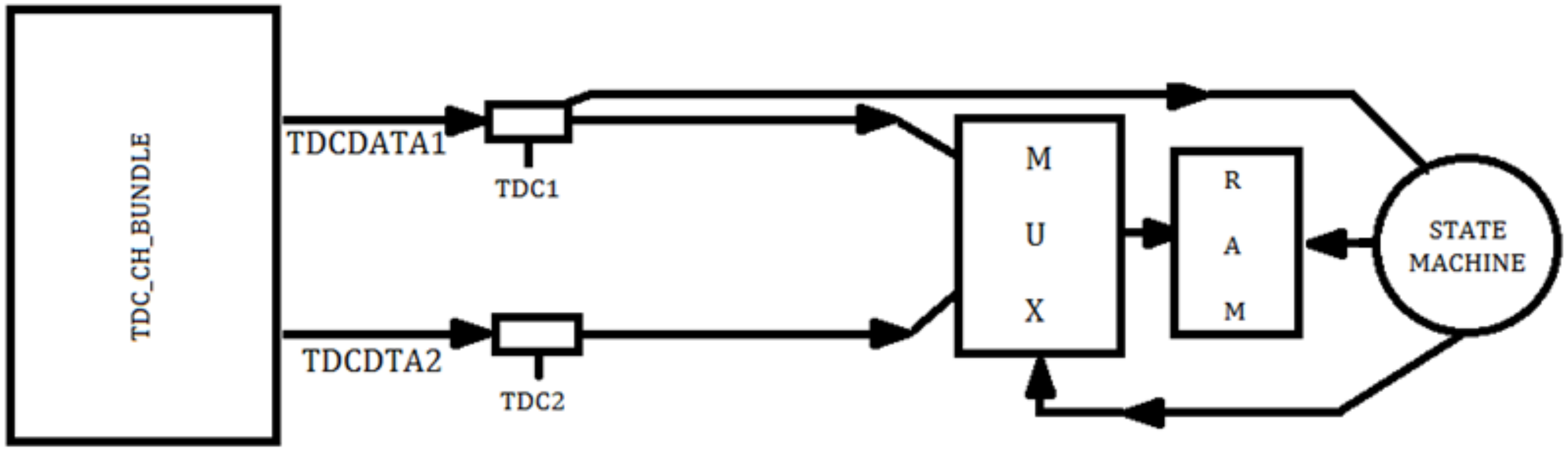


The system was debugged part by part and it worked, with the exception of the TDC_CH_BUNDLE. It was emulated and its emulation seemed to be working…so:

-the emulation was not sufficently accurate

- TDC_CH_BUNDLE really had a problem.

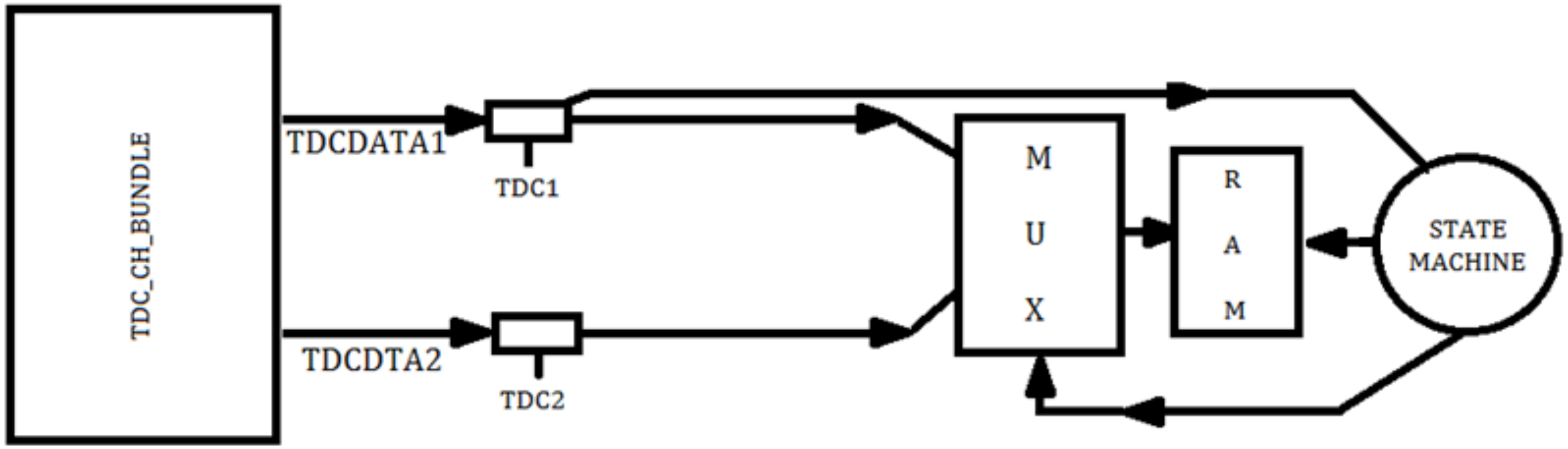
The answer…



TDC_CH_BUNDLE had problems with time constrains.

It has been sufficient to fix those problems to render the system completely working.

What to do now…



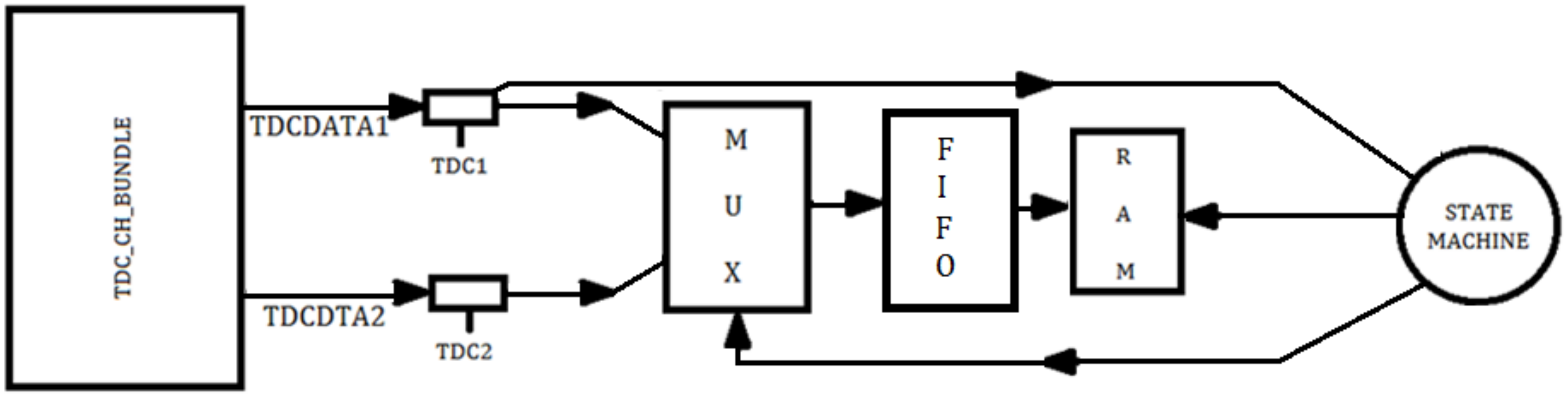
-The RAM is used only to debug the system.

-In place of the RAM, a device, called ROC, is used.

-This structure is not suitable to send data to ROC.

- ROC has to receive data from 16 indipendent
  channels. So, data reception has to be managed.

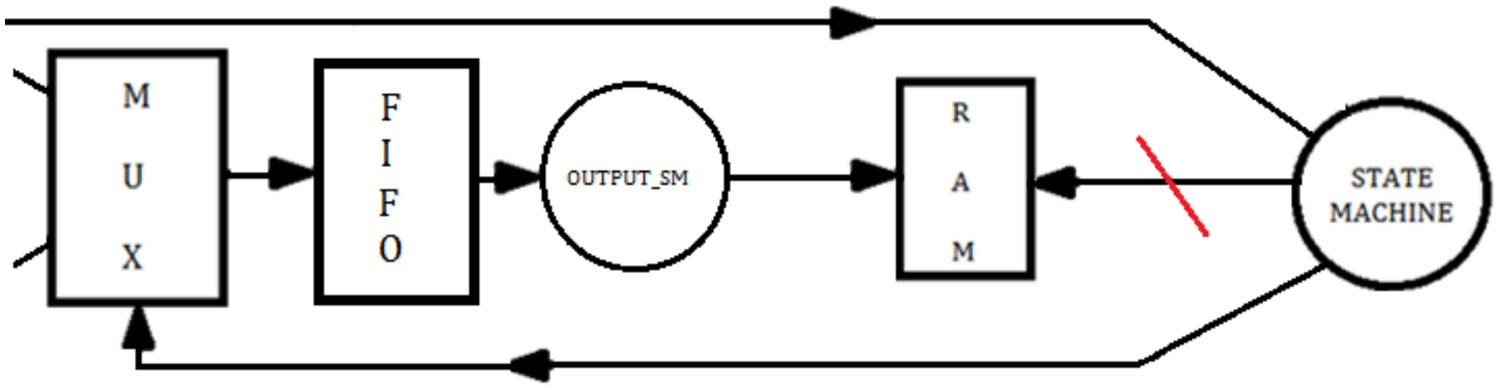
Inserting a FIFO



-FIFO (First In - First Out) is a structure which collects data so that the first –one inserted is also the first one which exits. Fifo is inserted after the MUX.

-The state machine has to be changed to send data to the FIFO with the correct temporization.

What drains the FIFO?

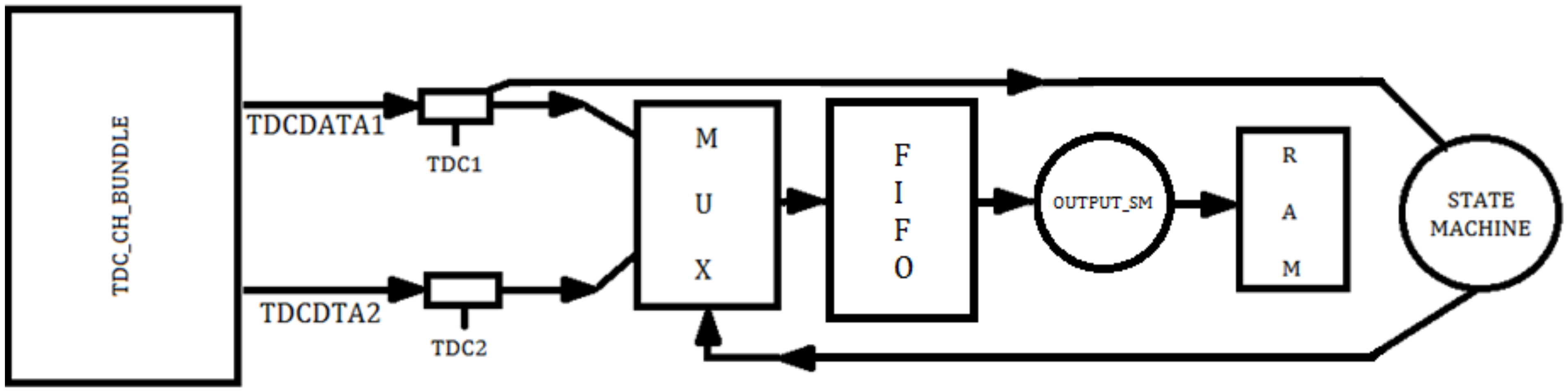


The OUTPUT_STATEMACHINE is used to drain the FIFO:

-The «old» state machine doesn’t manage the operation necessary to write into RAM anymore.

-FIFO can be read with a higher frequency than the one that is used to write into it -> it allows to avoid overflows.

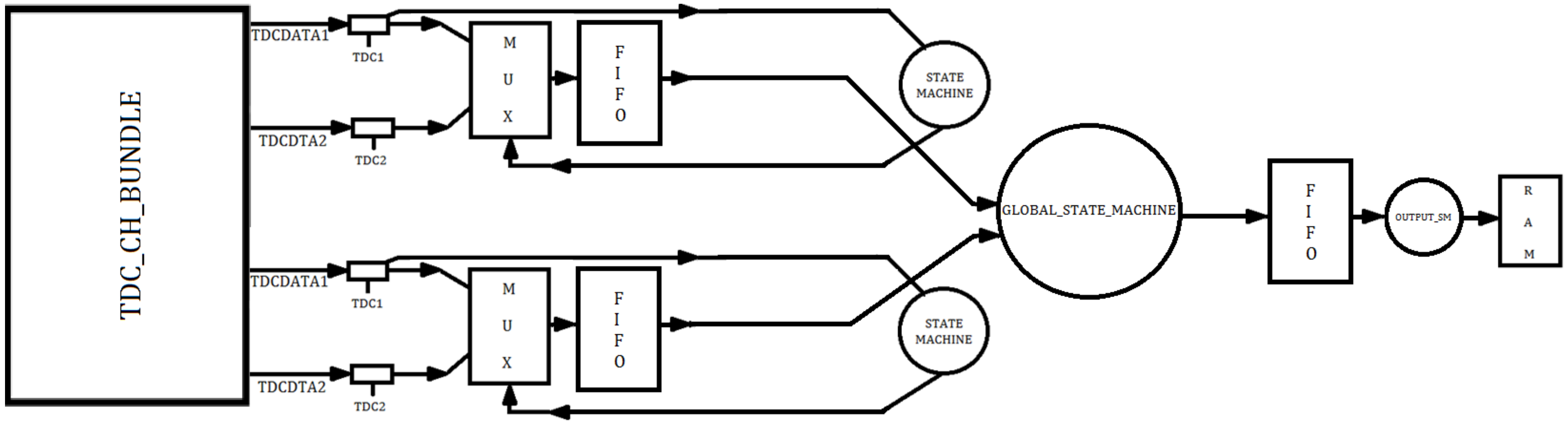
The channel's structure



This is the complete structure for each channel.

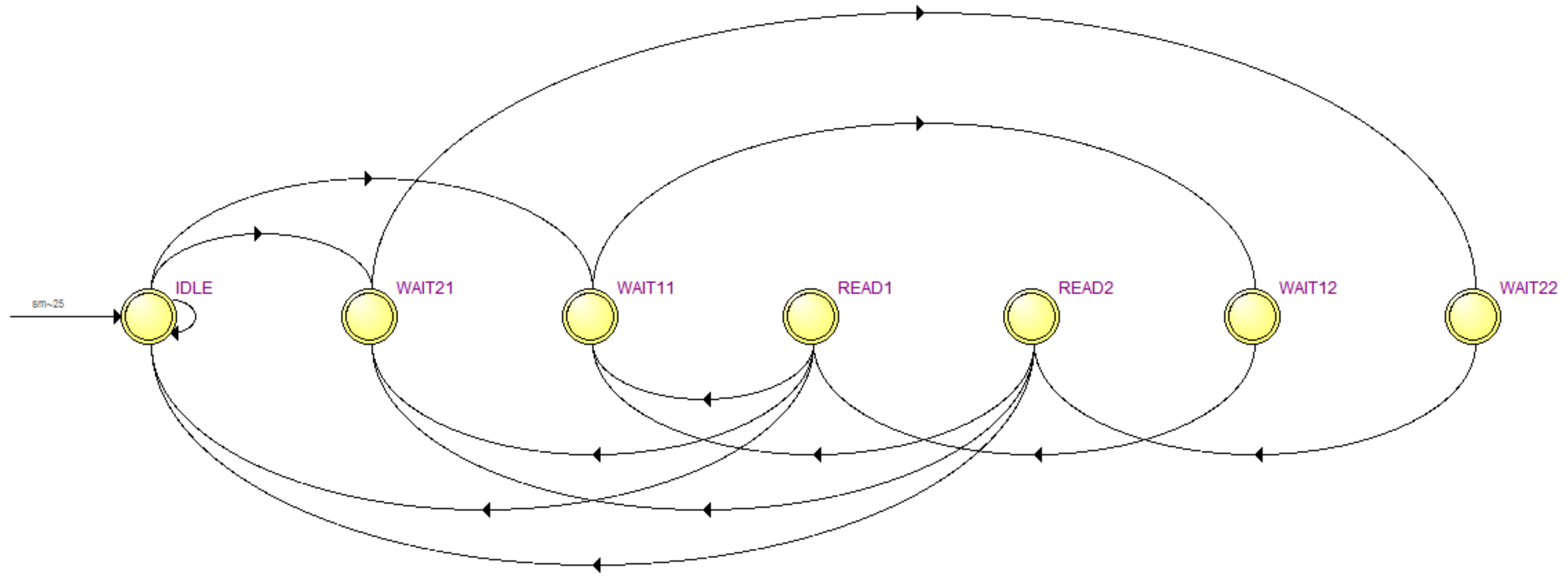
Anyway, this system is still not suitable to send data to ROC. In fact, a device, able to manage the data flow, coming from the different channels, is required. So, is not sufficient to copy this structure.

Two channels structure

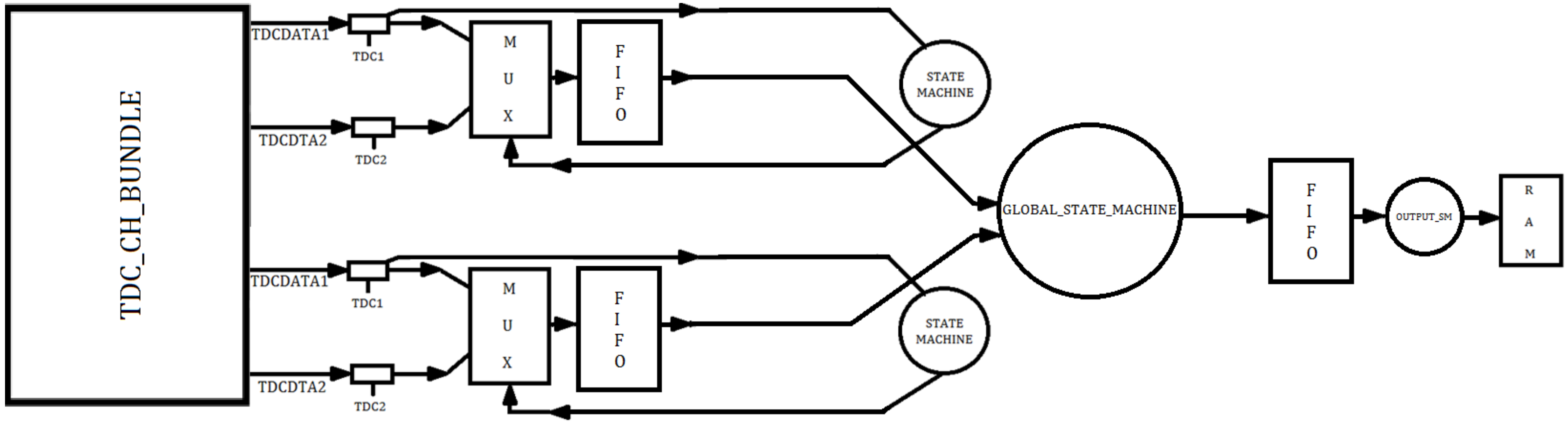


To solve this problem, a GLOBAL_STATE_MACHINE is inserted. This SM drains the FIFO of each channel and writes the data acquired into an other FIFO. This one is read by the OUTPUT_STATEMACHINE. RAM is still used only to debug the system.

The GLOBAL_SM’s state diagram



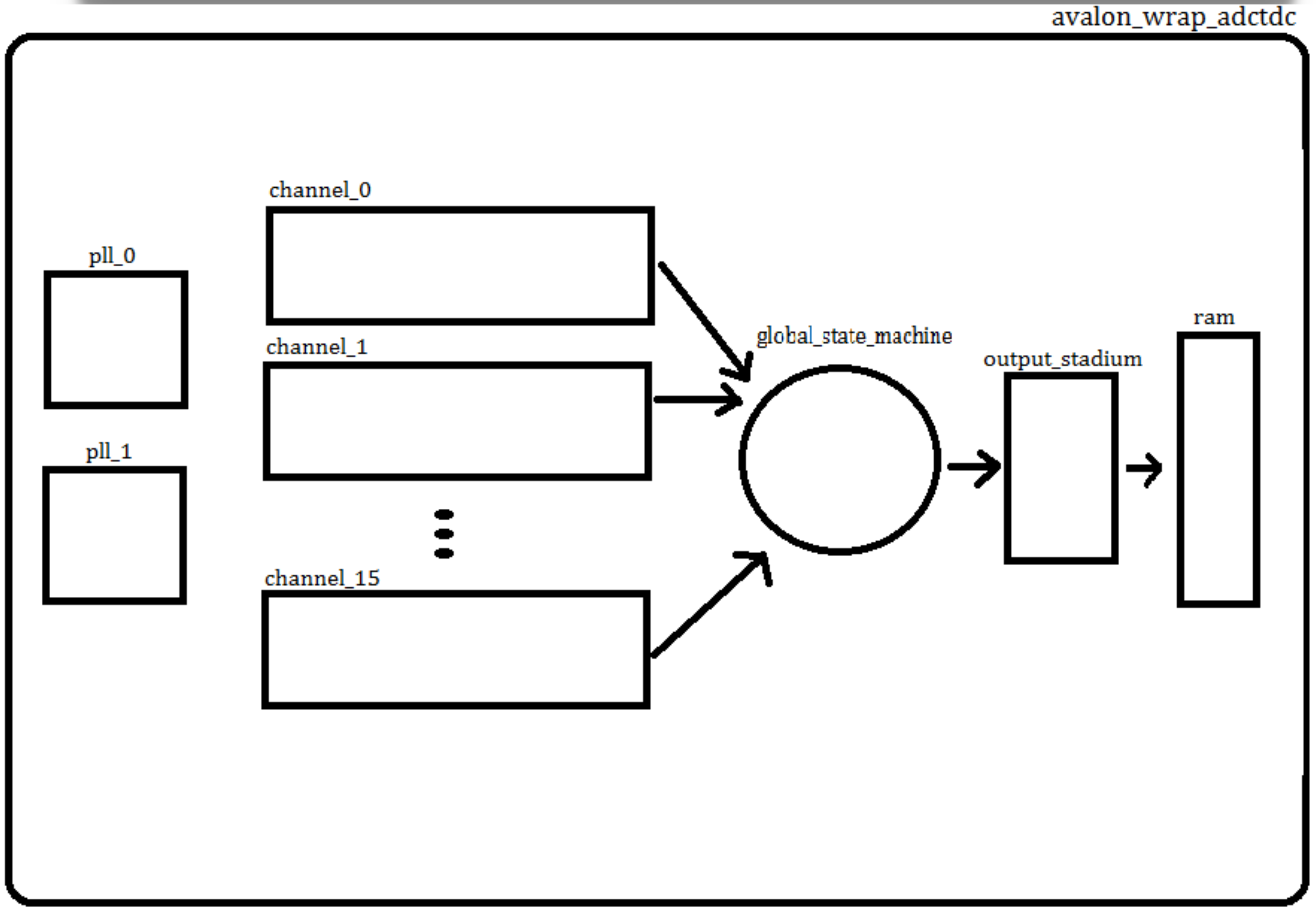
Almost ready…



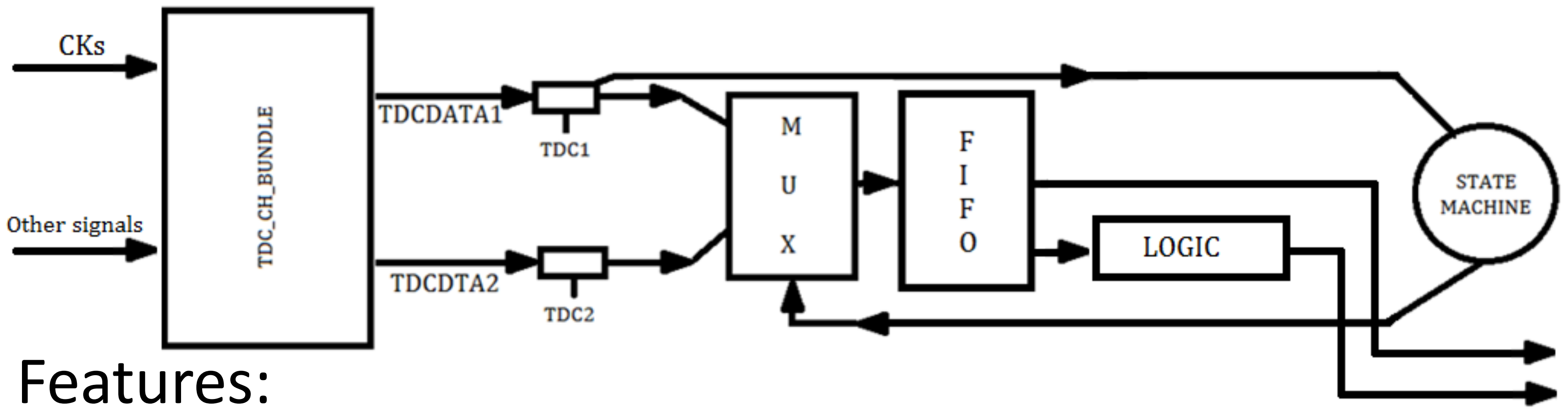
TDC_CH_BUNDLE managed two channels in the original project: (Martina took care about these problems)

- It has been optimized.
- It's necessary that each channel has its own TDC_CH_BUNDLE.

The complete system



Each channel's structure



Features:

-TDC_CH_BUNDLE indipendent for each channel.

-FIFO is used to collect data temporarily.

-CKs are common for each channel and they are provided by external PLLs, together with other signals to synchronize data.

Some differences…

According to the training program, the system should have been implemented on board of an IGLOO2 but we renounced. It was due to:

- Some problems occurred.

- The ROC has not been implemented yet, so it’s not so useful to change FPGA because data can’t be tested completely. We preferred to focus our attention on the system to optimize it.

Last days…

The system with two channels is working. So :

- We created a script in C++ to facilitate the debug

- We're trying to create a system with 16 channels. It's necessary to change the GLOBLAL_STATEMACHINE and render it «smarter» to avoid overflows in managing FIFOs.

Thanks for your attention!!!