ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Napoli

INFN/TC–97/04
30 Gennaio 1997

F. Fabozzi, P. Parascandolo:

OPERATION AND PERFORMANCES OF A HIGH SPEED DYNAMIC PATTERN DETECTOR

SIS–Pubblicazioni
dei Laboratori Nazionali di Frascati

INFN – Istituto Nazionale di Fisica Nucleare
Sezione di Napoli

INFN/TC–97/04
30 Gennaio 1997

OPERATION AND PERFORMANCES OF A HIGH SPEED DYNAMIC PATTERN DETECTOR

F. Fabozzi, P. Parascandolo
INFN – Sezione di Napoli, Mostra d'Oltremare, Pad. 20, I–80125 Napoli, Italy

Abstract

Performances of a TTL dynamic pattern detector for a high speed serial bit stream are reported.

Results show faithful operations up to a 12 nsec. clock period.

1. INTRODUCTION

The front end electronic cards (FEC) [1–2] of the IFR [3] sub–detector at BaBar [4] will be mounted at the edge of the RPC [5] detector. From this point the Data (50K channels) will be transmitted to 52 FIFO boards (IFB) located into 8 crates close to the apparatus which will provide for data buffering. Into these crates will house also ICB modules, whose function is to test the front end cards, and ITB modules providing the time occurrence of the hits.

Data acquired into the crates by the IFB and ITB must then be transmitted to the Data Acquisition system (DAQ).

The BaBar collaboration has chosen to multiplex a whole crate readout section into a single high speed transmission line going to standard DAQ Read Out Modules (ROM) common for all the sub–detectors and located in the Electronic House (EH). The approved solution for the transmission line consists of a fiber optic transmission system at a speed of 1.2 GHz (CLINK & DLINK). The fiber optic system serves both to acquire data from the detector to the DAQ (DLINK) and to transmit clock (59.5 MHz), trigger and control signals to the IFR sub–system crates (CLINK).

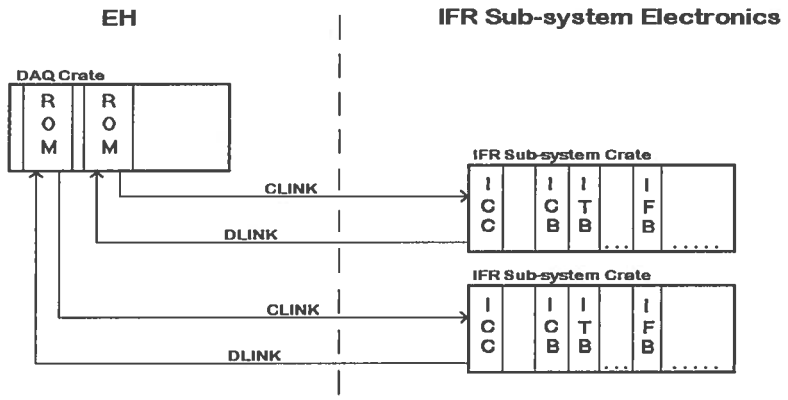


FIG. 1 – Overview of BaBar electronics.

This note first introduces to the established protocol between the standard ROM in the EH and the IFR sub–system; then, it focuses on how the serial bit–stream recognizing process is implemented in the IFB as this board is the most demanding one.

2. COMMAND FORMAT FROM THE ROM TO THE IFR SUB–SYSTEM

In the idle state the Data bit of the CLINK is stuck at zero while the 59.5MHz clock continues to run. That way the "time stamp" counters present on each sub–detector, and also employed on the IFB modules, keep up counting so that stay synchronous each other also when the CLINK is idle.

The serial bit patterns received by the IFR sub–system can be identifiable either as Run–time commands or as Non–run time commands. The former are in general time critical while the latter are those commands not allowed during data taking: set pattern mask, pulse width, etc.

A "command" is a serial string of twelve bit of Data (clock at 59.5 Mhz) the first one being always zero and the second one "start bit" always at 1.

The format of the Run–time commands is the following:

z s c c c c c d d d d d
z ==> 0
s ==> 1
c ==> command bits, LSB first
d ==> data bits, LSB first

Since 5 command bits are foreseen, a maximum of 32 Operation codes are allowed. The five Data bits are not sub–system definable. Commands and Data are transmitted with the LSB first.

On each module (IFB, ICB, ITB) a fixed pattern detector compares the serial bit stream against a predetermined pattern producing different check signals for each allowed pattern. For example, the IFB pattern detector compares the serial stream in order to produce the following commands: NOP, Clear Readout, Sync, L1 Trigger Accept, FC, Read Event.

NOP –code 0– is a no–operation code. With this command no operation is required onto the module. The command has to be decoded and a flag has to be written onto the FIFO (IFB module) with the next header.

Clear Readout –code 1– performs a reset operation of the event buffers. On the IFB module it performs a reset operation of the FIFO's pointers. That way the FIFO empty flag becomes active. This command is issued at the very beginning of the Data Acquisition or it is sent after a catastrophic event to restart acquisition.

Sync –code 2– is used to reset the time counters of the modules (for the IFB the "time stamp" counter referred to above).

L1 Trigger Accept –code 3– starts Data Acquisition from the front end. Each time this command is issued the strip status present on the detector 12 μsec. before will be latched onto the front end cards (FECs). As each FEC serves 16 channels, sixteen clock pulses are requested to transfer FEC's content serially to the IFB module serving 64 cards. In order to distinguish between successive triggers an Header (32 bit) and a Trailer (32 bit) must be added, so, for each L1 Trigger Accept, a total of eighteen write operations into the FIFO are requested. Since a FEC serves 16 channels, the DAQ –for each trigger issued– must read a total of 1088 bits from each IFB module. The data bits in the L1 Trigger Accept command constitute the "trigger tag" which is just a label to enable the DAQ system to distinguish between trigger frames.

Read Event –code 4– is used to start transmission of data held into the FIFO buffer to the DAQ. Data from the FIFO are first loaded into a shift register and then they are shifted out serially at 59.5 MHz with the LSB first.

False Command –code xx–. Each time there is no match between the serial input stream and the internal fixed pattern, the IFB sets a flag of False Command indicating a reception of a wrong pattern or a failure in the command decoding process. This flag is stored and written onto the FIFO with the next header.

Non–run time command have the format indicated below:

z s c c c c c a a a a a d d d .....
z ==> 0
s ==> 1
c ==> command bits, LSB first
a ==> sub–address bits, LSB first
d ==> data bits, LSB first

Bits z, s, c, c, c, c, c have the same meaning outlined above. The five "a" bits are sub–address bits (LSB first) to perform sub–command function inside a module. The data "d" pattern, always with the LSB bit first, consists of "n" bits and can be of variable length as necessary for sub–detector specific operation. In the IFR sub–detector: n=64 to write a test pattern into the IFB; but n=96 or 16 for operations onto the ICB module. In case of Non–run time read command there is no need for data bits so that n=0.

Non–run time commands are issued to the IFB module in order to test proper operation of the card. To accomplish this function the DAQ can send a WRF command –code 1A– to write a 64 bit pattern onto the FIFO and read back it via an RDF command –code 17–.

3. REQUIREMENTS IMPOSED ON THE RECEIVING AND ON THE TRANSMITTING SECTION OF THE IFR MODULES (IFB).

As outlined previously the ROM will operate according to the standard protocol of BaBar. This – in turn – fixes the following main interface specifications for the receiving section of the IFR modules [6]:

1) The incoming clock has a fixed frequency of 59.5 MHz.
2) The command protocol for both Run–time and Non–run time must adhere to the standard of Babar.
3) The jitter and phase stability of the CLINK base clock must be better than +/– 0.5 nsec.
4) The minimum time from the end of a command to the start bit of the next command is 15 nsec (one clock cycle) with the exceptions listed at the following points 5 and 6.
5) The minimum time between the start of transmission of subsequent L1 Trigger Accept commands must be 2.2 μsec.
6) The minimum time between an L1 Trigger Accept command and its Read event must be 2.2 μsec.
7) The minimum spacing between the end of transmission of a Trigger Accept command and the start bit of a following Read event command that belongs to an earlier issued Trigger Accept command is 15 nsec. (one clock cycle).
8) The ROM does not send Non–run time commands when in Run–time mode.

The requirements imposed on the transmitting section of the IFR modules (here IFB) are:

a) Idle state of transmission must be zero.
b) IFB must not transmit data unless requested by the ROM.
c) Header (32 bits for the IFB) must precede data and must be composed of an even number of bytes (one byte = 8 bits).
d) Trailer should contain an even number of bytes with all bits at zero (for the IFB 32 bits at one followed by 32 bits at zero).

4. PLD CHOICE

The design of a dynamic pattern detector operating continuously at a clock frequency of 59.5 MHz is rather tricky and requires a careful choice of the component since, to assure safe operating margins, the pattern detector should work up to 14 nsec. minimum, well beyond the 16 nsec operating clock period. To meet these timing specifications, the layout of printed circuit board needs to be very accurate and use of SMD components is a must.

To face the challenge of efficiently detecting the commands to this operating speed, two project strategies are possible: a State machine design or a shift & decode approach.

To implement the pattern decoder as a State machine a minimum of twelve States are requested. The flowchart should be:

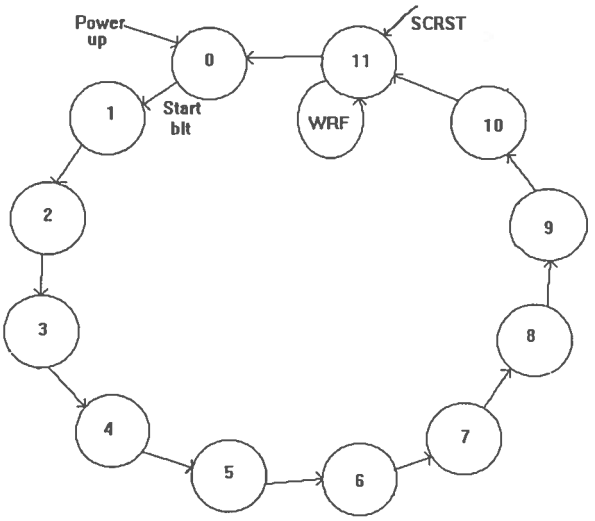


FIG. 2 – State machine flowchart.

As shown in the figure, whenever there is no activity on the Glink, the input of the PLD is stuck at zero and the State machine should be in the idle state, State 0. The transition to State 1 would take place when the start bit is decoded. The following transitions would take place synchronously with the 59.5 MHz clock up to State 11 where decoding of the pattern should be complete and, depending on the serial pattern received, one out of 8 command flag bit should be set. State 11 should act as the hub of the State machine. In fact transition from this state should depend on the detected command and should be toward State 0 for all but WRF command where it would request a wait time of 64 cycles into this State.

Although simple in principle, the use of a State machine at this operating speed can be tricky as the design goal is to go beyond a 14 nsec. clock period.

In general, when using a PLD, a first order estimation of the number of the product terms required is necessary to evaluate if a design fits. However counting the number of product terms generated by writing a file in a high level syntax is not straightforward. Often, having more than 12 product terms enforces the high level program to perform gate splitting using an extra feedback path. Clearly, gate splitting slows down the operating maximum frequency of operation. Furthermore, adding a 5 bit shift register and a pipeline register to latch the trigger tag plus some other necessary sparse logic, leads to a design in which gate splitting cannot be avoided especially for medium complexity PLD's.

5. THE COMPONENT

We have chosen to implement this design into a MACH210A. This component behaves as four PAL blocks each similar to a PAL22V16.

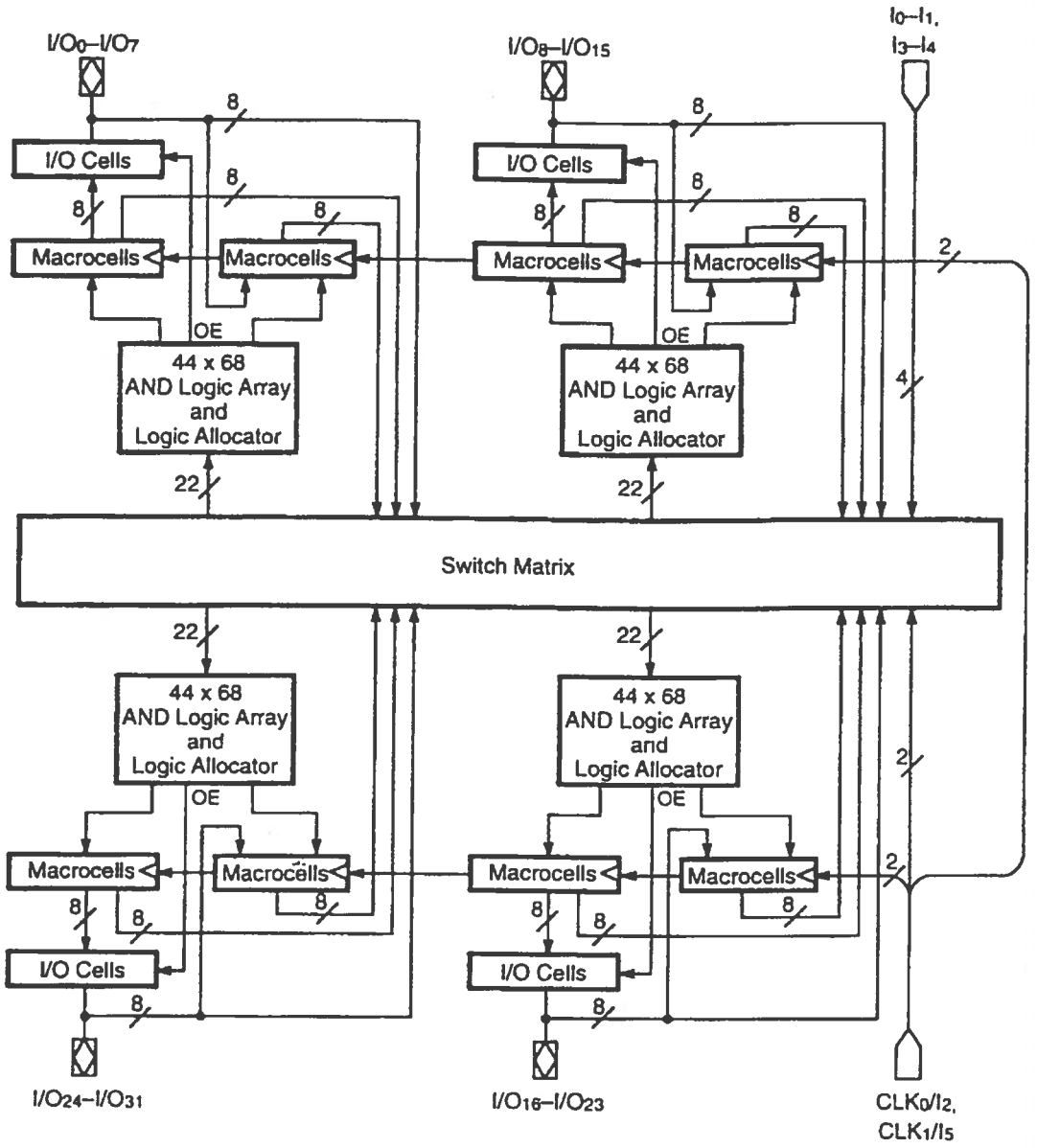


FIG. 3 – MACH210A.

For each PAL block there are 64 product terms available for logic use by the 16 macrocells 8 of which are buried. The other 8 macrocells feed the eight IO cells. Four special purposes product terms are also available. Two of these provide asynchronous preset and reset. Logic connections between the PAL blocks take place through feedback signals from the 16 macrocells and the eight IO cells via the switch matrix. The fastest version of this family of PLD's is the 7.5 nsec. version.

6. IMPLEMENTATION & ABEL FILE

To avoid the gate splitting process referred above, which could lead to a decrease in the maximum operating frequency, we have chosen a shift & decode approach. The pattern detector

is built around a 12 bit shift register Y[11:0] whose clock is CK60, the 59.5 MHz master clock. The outputs Y[6:0] of the shift register are used to identify commands.

Command flags stay active disabling all other commands until a reset is received.

Reset is produced after 12 clock cycles for all but the WRF command, where it arrives via the SCRST input after 12+64 cycles. Reset operation of the shift register must be performed in just one cycle to adhere to the BaBar protocol. Therefore, the asynchronous reset of the macrocells is not viable since the t_{AR} (asynchronous reset to output time) is 12 nsec. for the fastest device available in the family (MACH210A– 7). Consequently, to perform a reset of the shift register we have opted for a "write zero" operation which is much faster. The asynchronous reset is RS_OK signal produced by the PLD and externally connected to RST and is used just to reset not time critical registers such as the command flag register, delay registers, and strobe registers.

The two flag signals VC and FC are produced after 6 clock cycles and when active they disable the decoders so that two commands cannot be active at the same time.

GCD, GCDD and GCDDD build up a three stage delay to produce the RS_OK signal at the end of the twelfth bit of the string.

The "Trigger tag" buffering is implemented inside this PLD. On the twelfth cycle of an L1 Trigger Accept command a load signal (LD) is produced which provides the transfer of the Y[4:0] shift register bits into the pipeline register TAG[4:0]. Transfer takes place synchronously whenever LD signal is low. The same holds true for the "Time stamp" counter located into an external PLD. LD generation is therefore time critical and the relative equation cannot be combinatorial. To assure reliable operation the time t_{co} (delay clock to output) plus t_s (set up time from input, IO or feedback to clock) plus PCB delay (in case of "time stamp") must be well under our design goal (14 nsec).

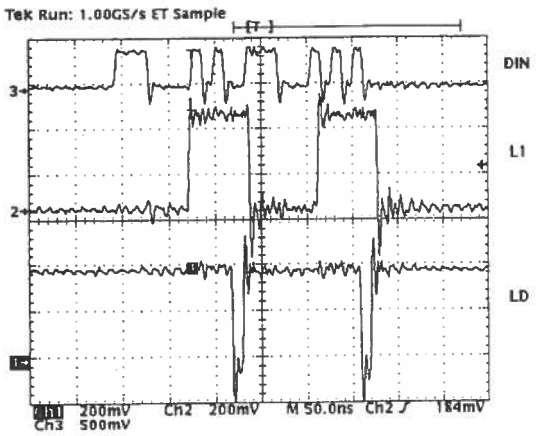
SLD output is active (low) whenever the detector recognizes a SYNC command and is used to synchronize the external "Time stamp" counter. For this signal the same rules as outlined for LD signal are applicable.

7. RESULTS

To assure good timing performance only D–type registers are employed as they have just a 5.5 nsec. setup time from input, IO or feedback to clock.

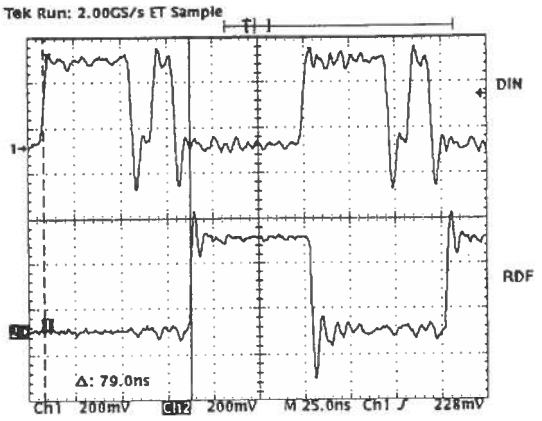
Obviously this setup time must be assured in the relative timing between the DATA IN input and the 59.5 MHz clock (CK60).

In the following graphs we show how, with a 5.5 nsec. setup time of DIN vs CK60, the PLD works with a 12 nsec period for some consecutive commands:
L1–L1, RDE–RDE, NOP–NOP, RDF–RDF.



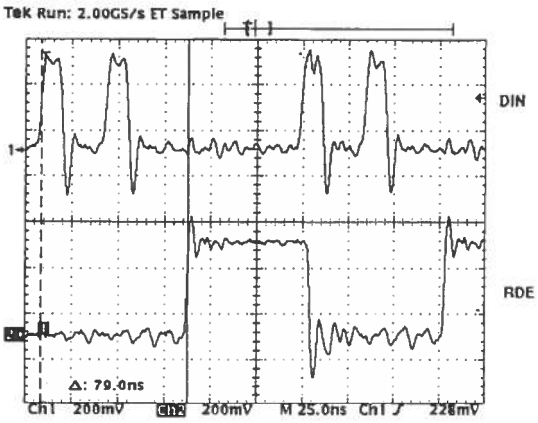
Two consecutive L1 commands

Clock period = 12ns



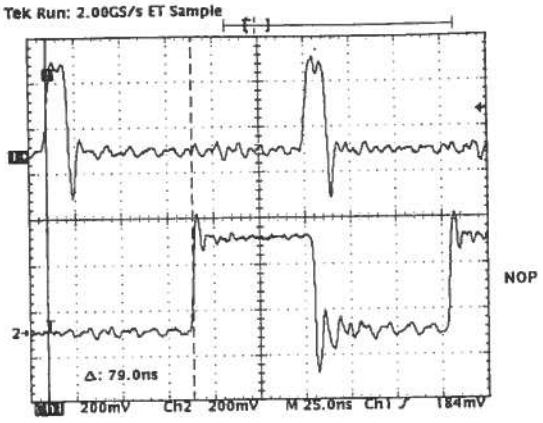
Two consecutive RDF commands

Clock period = 12ns



Two consecutive RDE commands

Clock period = 12ns



Two consecutine NOP commands

Clock period = 12ns

REFERENCES

[1] N. Cavallo et al., "A possible front–end readout scheme for the resistive plate chamber detector at BaBar", INFN/TC – 95/07.

[2] N. Cavallo et al., "Front–end card design for the RPC detector at BaBar", INFN/TC – 96/22.

[3] BaBar Technical Design Report, BaBar Collaboration, SLAC Report SLAC–R–95 457, March 1995.

[4] Letter of Intent for the Study of CP Violation and Heavy Flavour Physics at PEP II, BaBar Collaboration, SLAC Report SLAC–433, June 1994.

[5] R. Santonico and R. Cardarelli, Nucl. Instr. and Meth. **187**, 337 (1987).

[6] BaBar Note 281, BaBar DAQ Group, Draft 4/15/96.

APPENDIX

Abel file of the pattern recognizer (command decoder "CMDDEC")

```
module cmddec;
" November, 1996
title 'Command decoder';
cmddec device 'mach210a';

"————————————————— Version 2.0 ——————————————————————————
" // PWRUP signal, active high, provides reset to this PLD at start up.
" // VC signal (valid command) is active for all valid commands.
" // FC signal (false command) is active each time an unknown code is detected.
" // Valid and False command are treated the same way.
" // Three delay flip flop GCD, GCDD, GCDDD are used.
" // RS_OK equation to asynchronously reset the PLD is (!GCDDD & SCRST) # PWRUP.
" // For all but the WRF command RS_OK is produced by GCDDD. For the WRF command
" // RS_OK is generated by SCRST.
" // Shift register Y0–Y11 is reset by a "Write zero" operation.
" // "Trigger tag" buffered internally is put onto the FIFO input bus by the ENH signal.
" // LD is generated synchronously to load "trigger tag" & "time stamp".
"————————————————————————————————————————————————————————
"INPUTS
CLK, DIN, RST, PWRUP          pin 13,32,10,11;
SCRST, ENH                    pin 33,35;
"
" // CLK is the 59.5 MHz clock; DIN is the input data.
"
" ———————————————————————————————————————————————————————————
"OUTPUT
Y0, Y1, Y2, Y3, Y4            pin 17,18,19,20,21 istype 'reg_d, buffer';
Y5,Y6,Y7, Y8, Y9, Y10, Y11    node istype 'reg_d, buffer';
TAG0,TAG1,TAG2,TAG3,TAG4      pin 2,3,4,5,6 istype 'reg_d, buffer';
NOP, CLEAR, SYNC, L1, RDE     pin 37,38,8,41,40 istype 'reg_d, buffer';
WRF, RDF                      pin 42,43 istype 'reg_d, buffer';
NCLEAR                        pin 36 istype 'reg_d, invert';
FC,VC                         pin 27,26 istype 'reg_d, buffer';
WRF1, GCD, GCDD, GCDDD        node istype 'reg_d, buffer';
LD, RS_OK                     pin 24,25 istype 'reg_d, invert';
SLD                           pin 28 istype 'reg_d, invert';
"
" // Y11 – Y0 is a twelve bit shift register. Clock is CK60.
" // Y6 – Y0 are used to identify commands.
" // Commands stay active disabling all other commands until a "write zero" is made.
" // Node GCD is active one cycle after generations of VC and FC flags.
" // Whenever VC or FC are active, pattern decoders are disabled. That way two
" // commands cannot be active to the same time.
" // GCD, GCDD, GCDDD build up a three stage delay to produce RS_OK at the end of
" // the 12th bit of the string.
" // Y4 – Y0 on an L1 command contain the "trigger tag" and are transferred to the TAG register
" // whenever LD is active (low).
" // SLD is active on a SYNC command and synchronizes the external "time stamp" counter.
"————————————————————————————————————————————————————————

X = .x.;
C = .C.;
k = .k.;
Z = .Z.;
YY    =    [Y10,Y9,Y8,Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0,DIN];
KF    =    [RDE,L1,SYNC,CLEAR,NOP];
YT    =    [Y11..Y0];
TAG   =    [TAG4..TAG0];
```

```
TG     =       [Y4..Y0];

" // YY is the twelve bit vector input to the shift register.
" // YT is the twelve bit vector output of the shift register.
" // KF groups all but WRF and RDF commands.

equations
YT.clk =       CLK;
KF.clk =       CLK;
TAG.clk =      CLK;

[WRF.clk,RDF.clk,VC.clk,FC.clk,GCD.clk]                                  = CLK;
[WRF1.clk,GCD.clk,GCDD.clk,GCDDD.clk,LD.clk,RS_OK.clk,SLD.clk]   = CLK;

" // Macrocells have the same clock to assure synchronous operations
"-------------------------------------------------------------------------
[NOP.ar,CLEAR.ar,SYNC.ar,L1.ar,RDE.ar]                          = !RST;
[VC.ar,FC.ar,GCD.ar,GCDD.ar,GCDDD.ar]                          = !RST;
[LD.ar,RS_OK.ar,SLD.ar]                                        = !RST;
[WRF.ar,WRF1.ar,RDF.ar]                                        = !RST;
TAG.ar                                                         = PWRUP;
"-------------------------------------------------------------------------
TAG.OE                                                         = !ENH;
"
"
YT     :=      (YY & !(Y10&!WRF)) & SCRST ;
" // Write zero operation is performed whenever SCRST or Y10 are active
" // and command WRF is not valid.

NOP    := (!Y6 & Y5 & !Y4 & !Y3 & !Y2 & !Y1 & !Y0& !VC & !FC) #  NOP.fb;
CLEAR:= (!Y6 & Y5 & Y4 & !Y3 & !Y2 & !Y1 & !Y0 & !VC & !FC) # CLEAR.fb;
SYNC := ((!Y6 & Y5 & !Y4 & Y3 & !Y2 & !Y1 & !Y0 & !VC &!FC) #SYNC.fb)&!GCDDD;
L1     := (!Y6 & Y5 & Y4 & Y3 & !Y2 & !Y1 & !Y0 & !VC &!FC) #     L1.fb;
RDE    := (!Y6 & Y5 & !Y4 & !Y3 & Y2 & !Y1 & !Y0&!VC & !FC) #    RDE.fb;
WRF    := (!Y6 & Y5 & !Y4 & Y3 & !Y2 & Y1 & Y0 &!VC & !FC) #     WRF.fb;
RDF    := (!Y6 & Y5 & Y4 & Y3 & Y2 & !Y1 & Y0 &!VC & !FC) #      RDF.fb;

TAG    :=      (TG & (LD==0) # TAG.fb & (LD==1));

NCLEAR.clk =       CLK;
NCLEAR     :=      !CLEAR.fb;
NCLEAR.ar  =       !RST;

" // NCLEAR is produced to reset FIFO. This signal is normally high.
" // WRF1 is a delay stage on the WRF command so that GCD can be
" // correctly produced.

VC     :=      ( (NOP # CLEAR # SYNC # L1 # RDE # RDF # WRF) & Y6) # VC;
FC     :=      (!(NOP # CLEAR # SYNC # L1 # RDE # RDF # WRF) & Y6) # FC;
WRF1   :=      WRF;
GCD    :=      ((VC # FC) & !WRF1);
GCDD   :=      GCD;
GCDDD :=       GCDD;
RS_OK  :=     (!GCDDD & SCRST) # PWRUP;
LD     :=      ! ((L1) & !Y11 & !Y10 & GCDD);
SLD    :=      ! (SYNC.fb & !Y11 & !Y10 & GCDD);

end
```