

LSN- 158
O. D. Despe
November 7,1990

Arbitrary Function Generator for APS Injector Synchrotron Correction Magnets

Introduction

The APS injector synchrotron ring measures about 368 m in circumference. In order to obtain the precision of the magnetic field required for the positron acceleration from 450 Mev to 7.7 Gev with low beam loss, eighty correction magnets are distributed around its circumference. These magnets provide the vernier field changes required for beam orbit correction during the acceleration phase of the injector synchrotron cycle. Because of mechanical imperfections in the construction as well as installation of real dipole and multi-pole magnets, the exact field correction required at each correction magnet location is not known until a beam is actually accelerated. It is therefore essential that a means is provided to generate a correction field that is a function of the beam energy from injection until extraction for each correction magnet. The fairly large number of correction magnets in the system requires that the arbitrary function generator design be as simple as possible yet provide the required performance. An important, required performance feature is that the function can be changed or modified "on the fly" , to provide the operator with a real-time feel during the tune up process. The arbitrary function generator described here satisfies these requirements.

The arbitrary function generator (AFG) design is based on scanning out encoded data from a semi-conductor memory , a first-in-first-out (FIFO) device. The AFG data input consists of a maximum of 20 correction values randomly spaced within the injector synchrotron acceleration window. Additional points between these values are then linearly interpolated to create a uniformly spaced 1000 data-point function stored in the FIFO. Each point, encoded as a 2-bit value is scanned out in synchronism with the acceleration cycle and used to clock an up/down counter driving the digital-to-analog converter (DAC). The DAC in turn produces the analog reference voltage used by the current regulator control of the magnet supply, thus producing the arbitrary function correction field. Since the function data is first stored before use, its effect is a feedforward (open

loop) correction rather than a real-time feedback action. The optimum function for each correction magnet is determined via an iterative process based on the beam position during machine tune up.

1.0 Concepts Used In Function Generation

The design of the arbitrary function generator for the APS injector synchrotron correction magnets is based on the stair-case approximation of the line segments that represent the arbitrary function correction desired in the injector synchrotron acceleration period. In the APS power supply system the digital-to-analog converters (DACs) that generate the reference voltages to the current regulators are driven from the outputs of up/down counters. (See Fig. 1). The consequence of this design philosophy is that the control of a reference voltage (hence the output of a power supply) is made via pulse trains only.

Information derived from the points describing the arbitrary function, rather than the points themselves, are stored in a first-in-first-out (FIFO) memory. Each point is encoded in only a two bit value instead of the 11-bit value normally required for a number in the range of +/- 1000. A readout clock synchronized to the injection and extraction pulses of the injector synchrotron scans out data from each FIFO. The data are used to gate a clock to the up/down counter that drives a DAC. This enables the DAC to generate an analog signal whose effective slope can be controlled within some predetermined limits using a fixed clock rate and fixed step size. Circuitry is also provided that permits the "on the fly" update of the input function.

The simplest way to generate an arbitrary waveform to use as a reference in a system that uses an up/down counter-digital-to-analog converter (DAC) reference block is to feed pulses to the counter as in Fig 1. If the pulses can, in effect, be made to vary its rate, the DAC can produce any arbitrary waveform.

For simplicity, the clock is fixed in rate at its maximum value and lower rates are generated by gating off a fraction of the clock to the counter. Note that the maximum rate of change of analog output is limited to one least significant bit of the DAC per clock. If a higher analog rate is required, a higher clock rate is needed with a corresponding increase in DAC resolution requirement.

Fig. 2 shows an example of how a line segment AB can be approximated by incrementing or not at each clock time (using a fixed step size), a stair-case function.

The key information required is the state of the pulse enable (PEN) signal for each clock pulse in the acceleration interval. The pulse information at each clock time is stored in the FIFO. This data is then read out by a clock synchronized to the injection timing and pulses an UP/DOWN counter that drives the digital-to-analog converter (DAC) . The DAC produces the reference voltage used by the magnet current regulator. This basic configuration is depicted in Fig 3.

Between any two specified consecutive points a number of parameters are calculated to generate the linear interpolation values required; these are 1) the slope or count direction DIR, 2) the number of pulses between the two points and, 3) the average step count size between pulses. From these parameters the state of the pulse enable signal (PEN) at each clock time is calculated. The interpolation pulse information is three-valued: increment, decrement and, null. Fig. 4 shows the relevant parameters and some basic calculations.

2.0 Requirements

The requirements for the function generator are as follows:

- o Polarity Bipolar
- o Resolution (Amplitude & Time) 1×10^{-3}
 - 10 bits for magnitude
 - 1 bit for sign
- o Repetition Period 0.5 s
- o Function Duration (Acceleration) 0.25 s
- o Number of points specified ≤ 20
- o Number of points supplied 1000
(Linear Interpolation used between specified points)
- o Dynamic update of arbitrary function

3.0 Basic Implementation

Since the acceleration interval is divided into 1000 points and the function duration is 250 ms, the required clock period is 250 us. The points that define the function to be generated (the default function is 0 DC) are input to the Power Supply Control Unit software where the Pulse ENable states, the DIR signal, and other initialization parameters are determined prior to loading the FIFO.

The pulse enable states together with the direction information require a 2-bit wide, 1001-bit deep first-in-first-out (FIFO) memory. Fig 3 shows the basic idea of the hardware. A 4kHz clock synched to the injection pulse scans out the FIFO contents. The clock is gated by the enable signal and steered by the direction signal to UP or Down inputs of the counter feeding the reference DAC. After beam extraction the FIFO read pointer is returned to location 0 to prepare for the next scan cycle.

4.0 Eight Channel Implementation

As many as eight correction magnet supplies are controlled by one power supply control unit (PSCU). (The PSCU contains the intelligence that interfaces directly to each power supply hardware for control, monitoring and communicating to the host computer). The discussion above dealt with the concept of the correction magnet function generator. This section will discuss some details of the real implementation.

Fig. 5 shows the correction magnet timing in an injector synchrotron cycle. For the purpose of this discussion, the injection pulse J is the reference pulse. The extraction pulse X is at the midpoint of the injector synchrotron cycle. The FIFO empty flag signal is generated when the read pointer equals the write pointer. This should occur one clock pulse after X when the DAC and the FIFO read pointer are reset to zero. The interval between the empty flag signal and the following injection signal is reserved for returning the DAC output to its DC bias level and the update of FIFO data if an update request signal is present.

The high level block diagram of the full implementation is shown in Fig. 6. The main function blocks as indicated in the figure are given below with a brief description of each block.

The update of the DC bias level requires that the number of pulses programmed into the unit-gated-count-generator (UGCG) is changed to the new value before the empty flag signal occurs which enables the UGCG that ramps the DAC to its DC level. The appropriate time window to do this is between injection J and the empty flag signal so that new UGCG data is ready to drive the DAC after reset. The FIFO data update also occurs between the empty flag signal and before the next injection pulse J. Note that all signals from the main section block with the exception of the FIFO ID select signal are common to all eight PS FIFO blocks.

Main FIFO-

This provides temporary storage of arbitrary function data before loading the addressed power supply FIFO. This offloads the main CPU, as a hardware clock is used to read this FIFO. It also eases synchronization and programming.

Power Supply FIFO (8)-

Each FIFO stores arbitrary function data for its power supply. It can receive update data from main FIFO and provides sync signal for the update process from its empty flag. Since all PS FIFOs are scanned in synchronism with the injection pulse, only one empty flag signal is used.

Main Gated Count Generator (MGCG)-

This supplies the required number of clock pulses (20kHz) to load the addressed PS FIFO from Main FIFO. It is gated on by presence of update request signal at time PS FIFO becomes empty. This frequency is chosen to enable the PS FIFO update process to be completed well within the time window available for loading the FIFO.

Unit Gated Count Generator (UGCG) (8)-

This unit generates the required number of pulses (100kHz) required to reset the DAC value to its DC bias level after every beam extraction time. On an update request, this unit is reprogrammed with the new DC level to use with the new FIFO data within the time window indicated in Fig. 5.

Logic Support-

This block generates various other signals needed to synchronize the sequences of events and select clock sources.

Programmable Logic Device (PLD)-

The PLD is used to implement combinational/sequential logic of nine inputs consisting of data, clock sources, timing and control signals in a single 24-pin dual inline package.

5.0 Operation of the Arbitrary Function Generator

The operation of the arbitrary function generator can be divided into the following operational phases after software has determined the PEN, DIR and other related signals. Each phase is described and a pseudo code given. The phases are not necessarily mutually exclusive. The loading of the Main FIFO, for example, is completely asynchronous to and independent of any PS FIFO operation.

Processing/Loading of Arbitrary Function Data Points-

The Power Supply Control Unit (PSCU) receives from its host computer the data points describing the arbitrary function. The PSCU maintains a table of the latest arbitrary function data, (20 points) for each power supply it services. From these data it makes calculations of the slope (DIR) and pulse enable signals (PEN) associated with the FIFO readout pulses for each line segment. In order to save memory, the newly obtained DIR and PEN signals are immediately clocked into the main FIFO. This calculate-load cycle is repeated until 1001 points (1000 for acceleration, 1 to generate empty flag signal) are stored in the main FIFO.

The software also determines the number of pulses required to set the DAC counter to a number corresponding to the DC bias for the new function. This number is programmed into the gated count generator (UGCG) immediately after injection if an update request is outstanding. The software generates update request and ID select signals to address the specific FIFO involved.

Pseudo code for above process after parameter calculation:

```
Reset main FIFO read and write pointers;
Repeat until all data loaded to main FIFO
{ present Input Data to main FIFO;
  clock in main FIFO;
}
Generate Update-Request signal;
wait until start of injection
program UGCG;
issue ID select signal;
```

Reading Main FIFO and Loading PS FIFO-

On the next empty flag signal after the latched update request, the addressed FIFO is reset and the unit gated count generator (UGCG) is activated delivering the programmed count number of pulses (100kHz) to the selected DAC counter to set it to the new DC level. At the same time, the main gated count generator (MGCG) is started, reads out the main FIFO and loads the addressed FIFO. The update request latch is then reset. The new function is therefore in place when the next injection pulse occurs.

The following pseudo code describes the monitoring of update request and transfer of main FIFO data to unit FIFO:

```

Repeat forever
  (if (At time PS FIFO flag empty and update_request true)

      { Reset pointers of selected FIFO;
        Reset its DAC to zero;
        Enable Main Gated Count Generator;
        Clock in data to addressed FIFO;
        Enable selected UGCG to counter;
        Reset Update_request flag;
      }
  )
}

```

Scanning PS FIFO Data -

In its normal mode, each function generator is repetitively scanning its FIFO in synchronism with the acceleration cycle. At every injection pulse, the 4kHz clock is selected to scan the FIFO and drive the DAC counter in accordance with the unit's direction UDIR and unit's pulse enable UPEN signals. At each empty flag occurrence, the DAC counter and the read pointer of the addressed FIFO are reset to zero. The UGCG is then enabled and sets the counter to its DC bias value in preparation for the next injection.

If at injection time an update request is present, the following additional actions are performed within the injector synchrotron cycle: 1) injection timing signal is sent to the microprocessor for synchronization, 2) the UGCG is programmed with new DC bias data, and 3) addressed FIFO is loaded with new data. These actions are transparent to the normal mode of operation of the function generator as described above. The programmable logic device PLD handles much of the clock selection, routing and gating of signals to the FIFO and the DAC.

The modes of operation described are illustrated by pseudo codes. Note that they are concurrently executing within an injector synchrotron cycle.

Normal operational cycle:

```
repeat forever
{ if (injection signal detected)
  { select 4kHz clock;
    scan FIFO;
    drive DAC;

    if (empty flag detected)
    { Reset FIFO read pointer;
      Reset DAC Counter;
      Enable UGCG to set DAC DC bias;
    }
  }
}
```

Update Request cycle:

```
if at injection (update request present)
{   signal microprocessor injection time;
    program UGCG;

    At Empty Flag time
    {Reset addressed FIFO;
     reset DAC counter;
     Enable UGCG to set DAC new DC bias;
     read out main FIFO and load addressed FIFO;
    }
}
```

6.0 Reference

1) Kuninori Endo, "Minicomputer Assisted Function Generator for the Dynamic Control of Chromaticity Correction Sextupoles", IEEE Transactions on Nuclear Science, Vol. NS-26 No. 3, June 1979

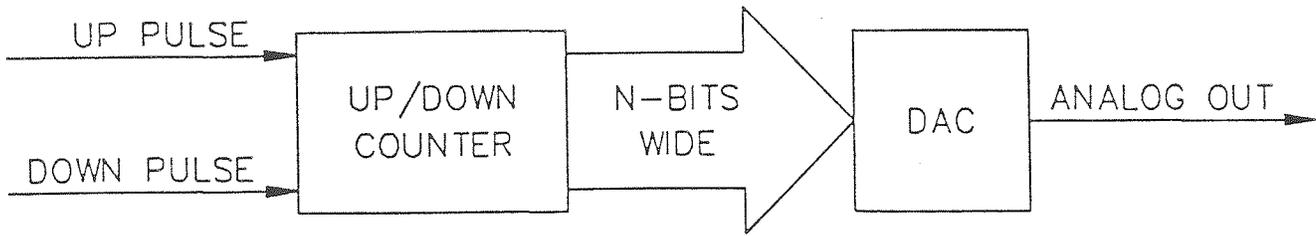


FIG. 1 ANALOG REFERENCE BLOCK

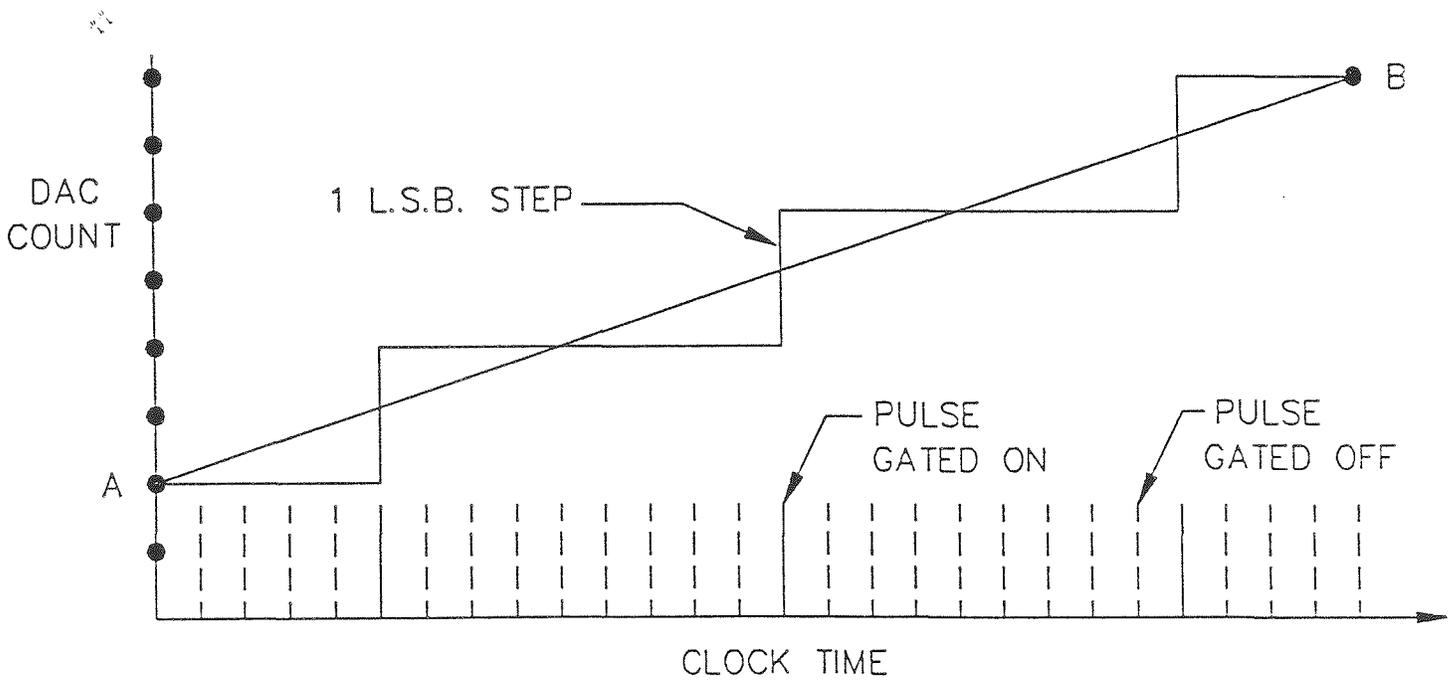


FIG. 2 STAIRCASE LINE APPROXIMATION

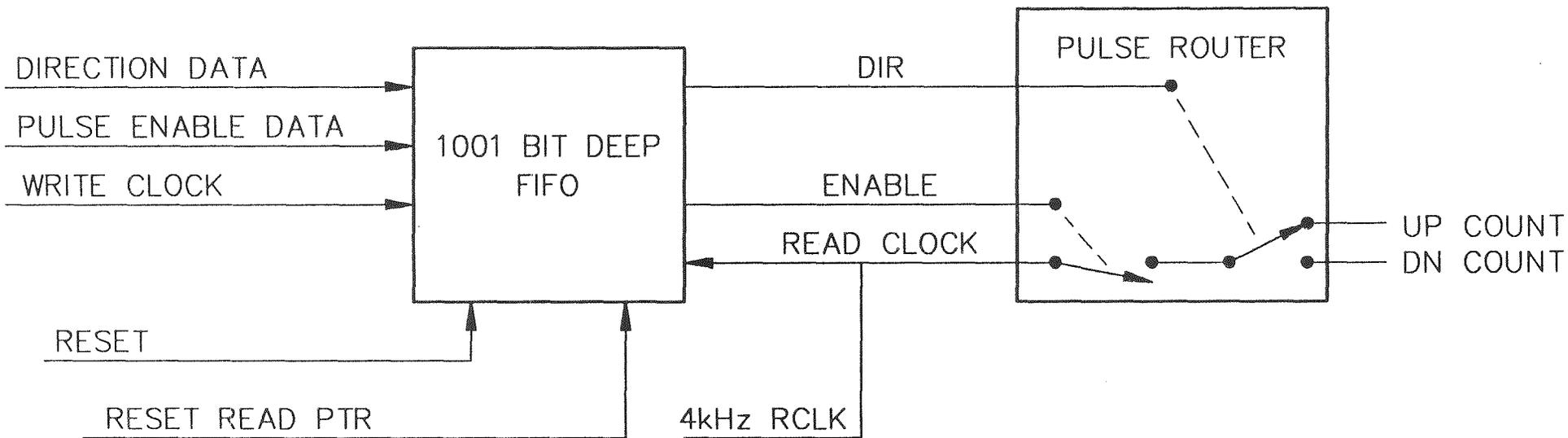


FIG. 3 FIFO CONFIGURATION

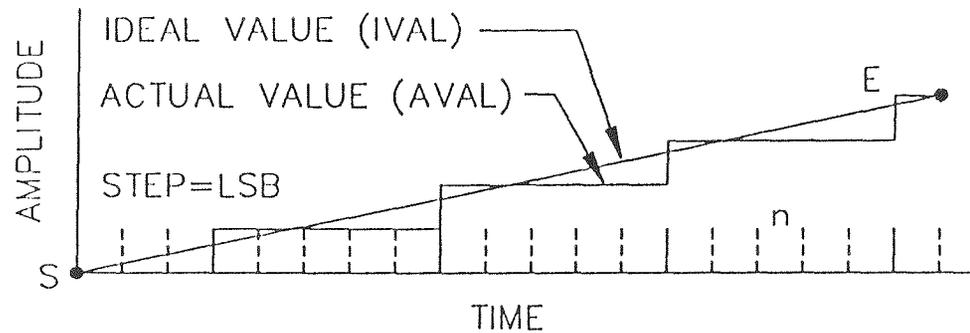


FIG. 4 STAIRCASE APPROXIMATION TO A LINE SEGMENT

| PEN=1 PULSE ENABLED
 | PEN=0 PULSE DISABLED
 n nth CLOCK TICK FROM S

FROM FIG. 4 THE FOLLOWING RELATIONS ARE ESTABLISHED:

SLOPE $M = (E-S)/N$ $N =$ NUMBER OF PULSES BETWEEN S & E

DIFFERENCE = $ABS(IVAL - AVAL)$

SIGN = $ABS(E-S)/(E-S)$ (+ -1)

ALGORITHM FOR DETERMINING PULSE ENABLE (PEN) SIGNAL:

IF VALUE OF DIFFERENCE IS $>$ HALF LEAST SIGNIFICANT BIT,

THEN

ENABLE PEN=1

ELSE

DISABLE PEN=0

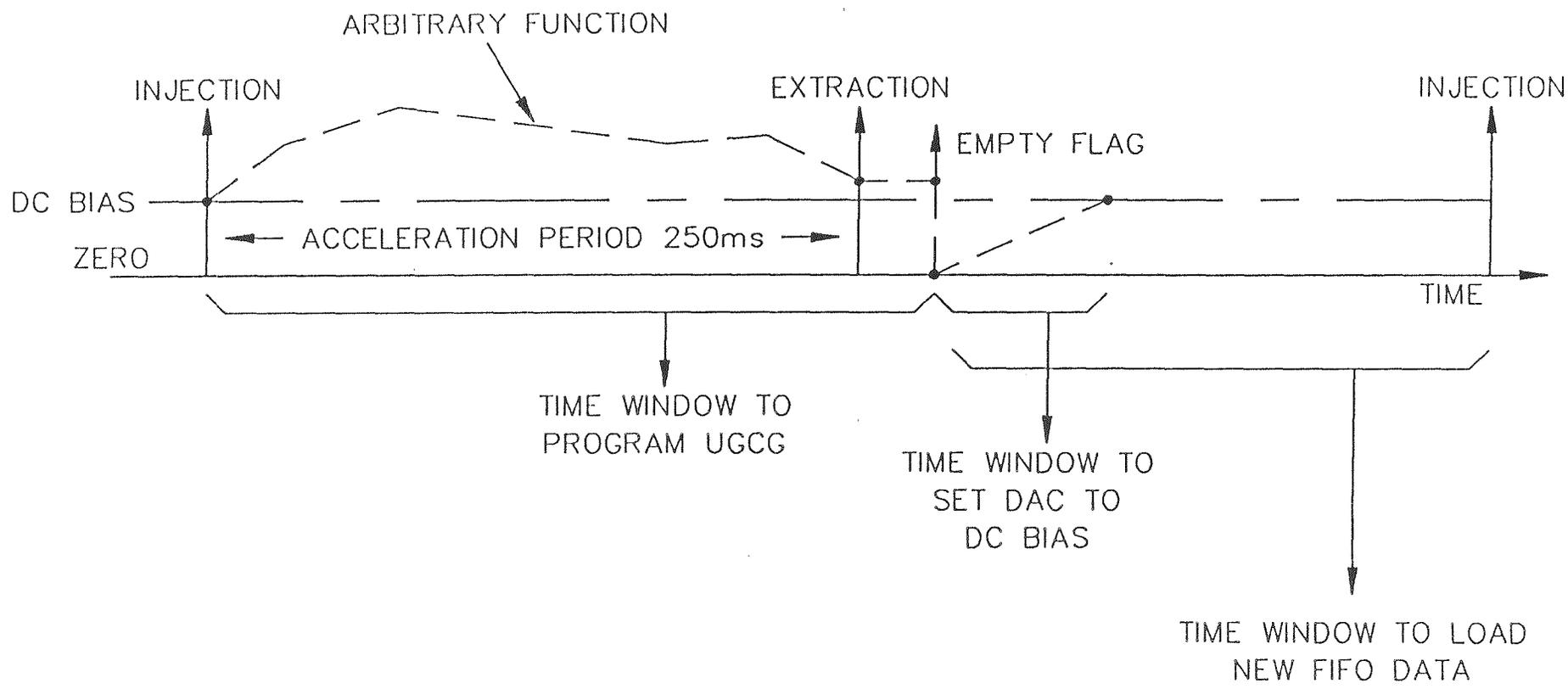


FIG. 5 ARBITRARY FUNCTION TIMING IN A SYNCHROTRON CYCLE

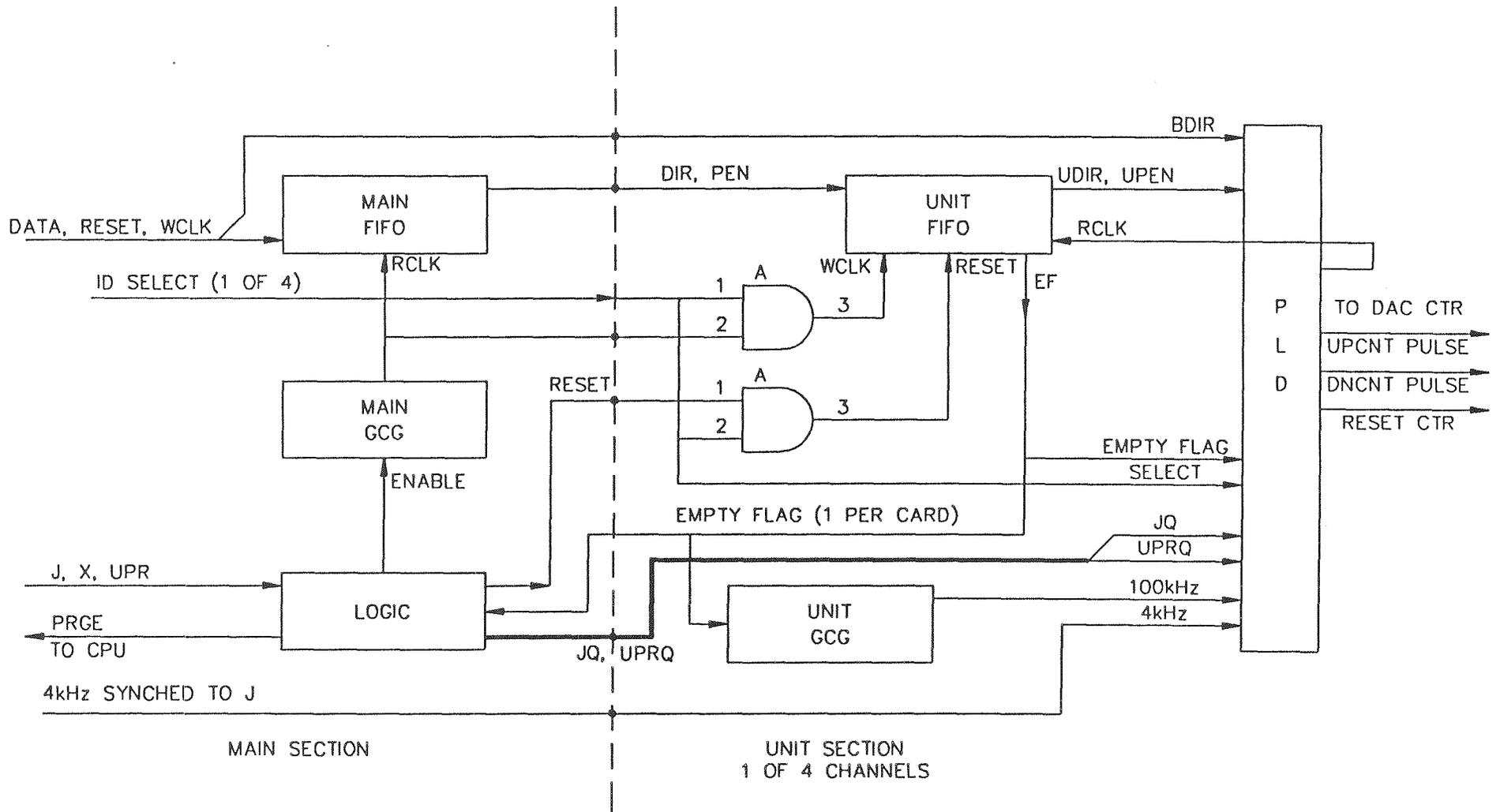


FIG. 6 ARBITRARY FUNCTION GENERATOR BLOCK DIAGRAM