Prototype Description

Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

DVB-T2 Receiver Prototype: Physical Layer

Jan Kloub, Tomáš Mazanec, Antonín Heřmánek, Milan Tichý
{*kloub,mazanec,hermanek,tichy*}*@utia.cas.cz*
+420-2-6605 2212

Contents

© 2009 UTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved.

Revisions

Revision	Date	Author	Description of document changes
1	Aug 27, 2009	MT	Document creation
2	Sep 23, 2009	MT	Major revision; complete rewrite
3			
4			
5			

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

1 Introduction

This document describes the physical layer of the DVB-T2 receiver prototype.[1] The prototype is a product of cooperation between UTIA AS CR (Czech republic) and Screen Service (Italy). The aim of the project is to built a professional DVB-T2 receiver for signal measurements and testing of the DVB-T2 broadcasting system, the second generation digital terrestrial television broadcasting system that is standardized by the European telecommunication standard ETSI EN 302 755 [1].

The DVB-T2 receiver implementation consists of two levels of signal processing. The first level contains P1 processing implementation, where P1 symbol is detected and decoded within received T2 frame. The second level contains P2 and data symbols processing. The P2 can be properly processed when all required parameters of T2 frame are known. The FFT size is determined by the decoded P1 symbol. Next, the guard interval of P2/data symbols together with the number of symbols in T2 frame are determined. Finally, the time synchronization of T2 frame symbols is performed. When all frame parameters are known, the P2 processing can start.

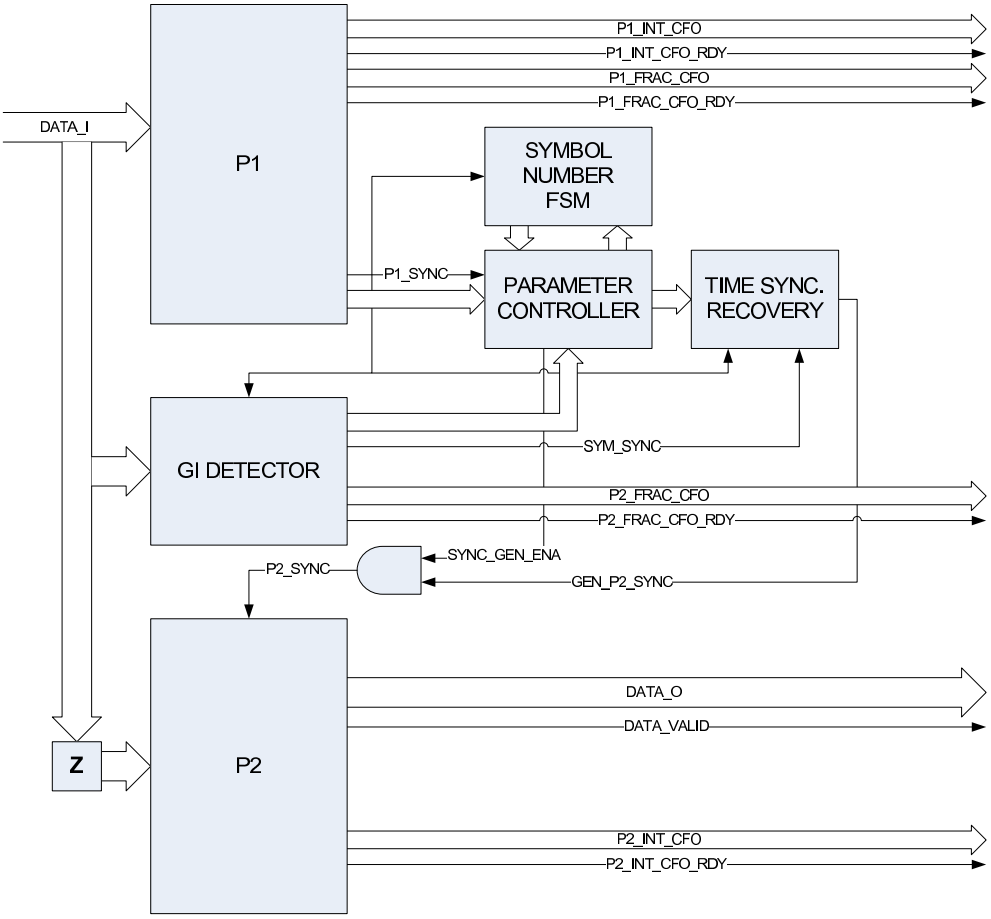

Figure 1: Top level of DVB-T2 receiver block diagram

P1 processing is described in Section 2, guard interval detector is described in Section 3, determining the count of frame symbols is described in Section 5, fine time synchronization is described in Section 6 and P2 processing is described in Section 7. The top level of DVB-T2 receiver block diagram is shown in Figure 1.

[1] Since information on the DVB-T2 receiver prototype is confidential and the prototype is in exclusive ownership of the Screen Service Broadcasting Technologies, this document contains only a brief description of the DVB-T2 receiver prototype implementation. The full documentation has been provided to the Screen Service Broadcasting Technologies in [3].

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

2 P1 Processing Implementation

The implementation of the P1 symbol decoder consists of several steps. Each step is represented as a functional block in VHDL design. The first step is to find data of the P1 symbol to be decoded. Continuously, incoming data are correlated for the P1 symbol position detection as described in Section 2.1. The P1 symbol is detected when peak value of correlator output is detected. The maximum detection implementation is described in Section 2.2. The P1 symbol is detected after the whole symbol has been correlated. The input data are continuously stored in a delay link implemented as circular buffer. The buffer overlaps difference between the P1 detection position and P1 symbol data position. The buffer output is used for the P1 symbol data decoding. The decoding method is described in Section 2.3. Storing the output data is controlled by finite state machine (FSM) which starts storing after the P1 symbol has been detected. The P1 decoder basic block diagram is depicted in Figure 2.

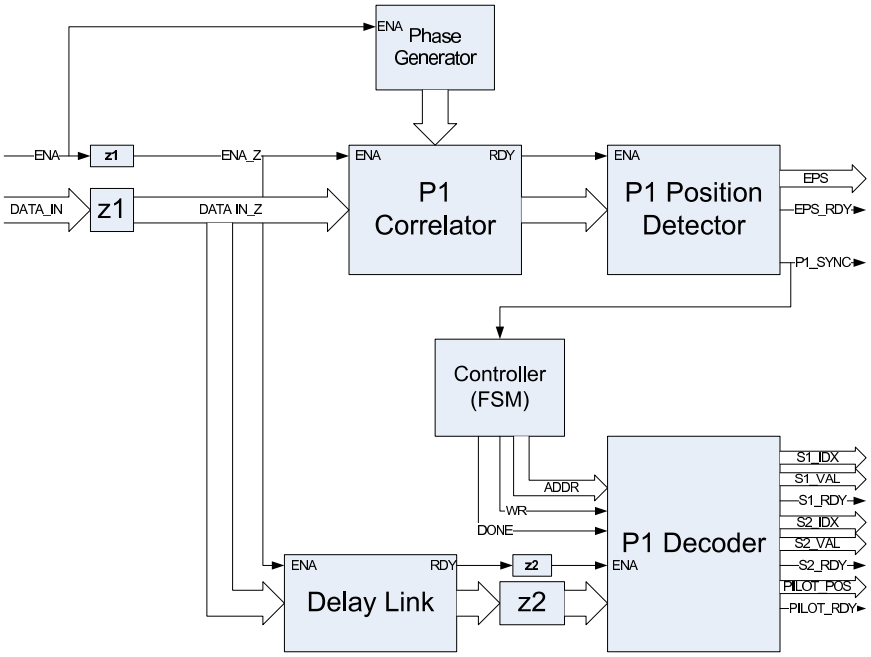

Figure 2: P1 processing unit implementation block diagram

2.1 P1 Symbol Correlation

The P1 correlator was implemented using the method described in the DVB-T2 Implementation Guidelines [2]. The P1 correlator hardware module implementation is shown in Figure 3. To keep all data path in the same word width scaling is necessary. Scaling is done using simplified divider which divides only with power of two. The running average is implemented only as a sum and scaling component replaces average division. Due to different data paths latencies the compensation is necessary. The compensation and delay components are realized using shift registers or circular buffer, respectively.

2.2 P1 Symbol Position Detection

The maximal magnitude position of P1 correlator output must be detected to find the P1 symbol data to decode. The magnitude of the correlator output is determined using CORDIC module. The correlator output data have to be scaled to keep consistent data path width in the design. As the correlator

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

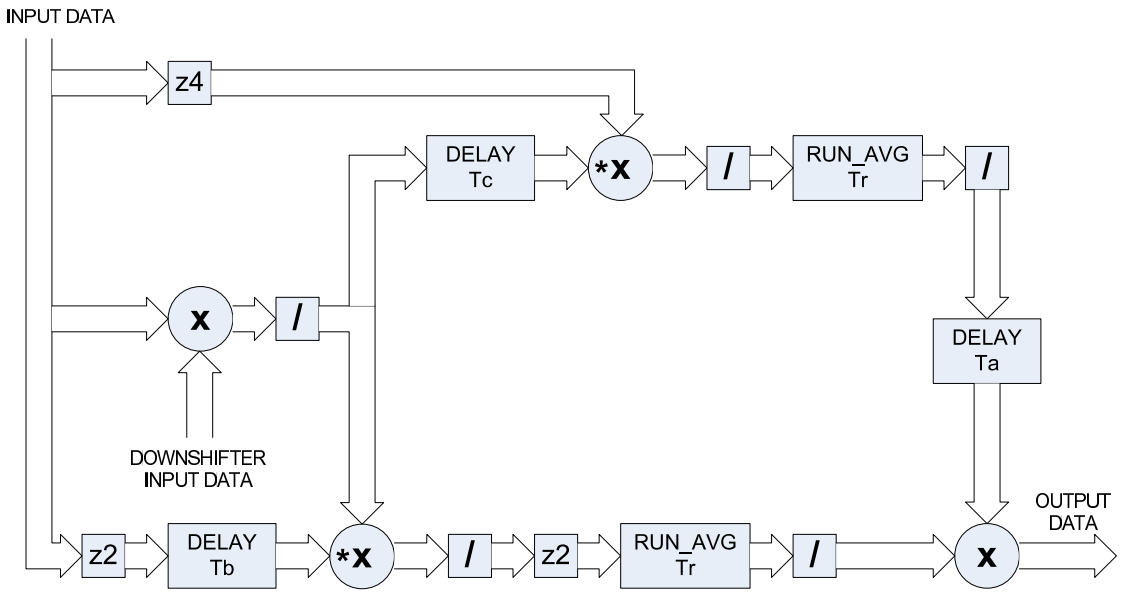

Figure 3: P1 detection correlator block diagram

simulation shows, to design data path width of the correlator output the division is not suitable—scaling with saturation was used instead. The scaling component is shown in Figure 4 as box with "**S**" symbol.

The P1 correlator output magnitude generates a peak when P1 symbol is detected. The adaptive threshold estimation have to be performed to detect the start of the peak. Due to high variance of the estimated threshold value the threshold estimator is filtered. For filtering, exponential forgetting was used.

When start of the peak is detected by comparator the search for maximum is started. The peak start is detected by the edge detector. The structure of edge detector is shown in Figure 4 as box with "**B**" symbol. The finite state machine starts waiting for maximum position. The maximum detection is based on gradient method. The P1 peak waveform has a plateau and thus gradient method could detect several peak positions while plateau is observed. Only the first peak detection is valid and the rest peak detections are invalidated for several (P1 symbol length) data samples to mitigate the effect of plateau.

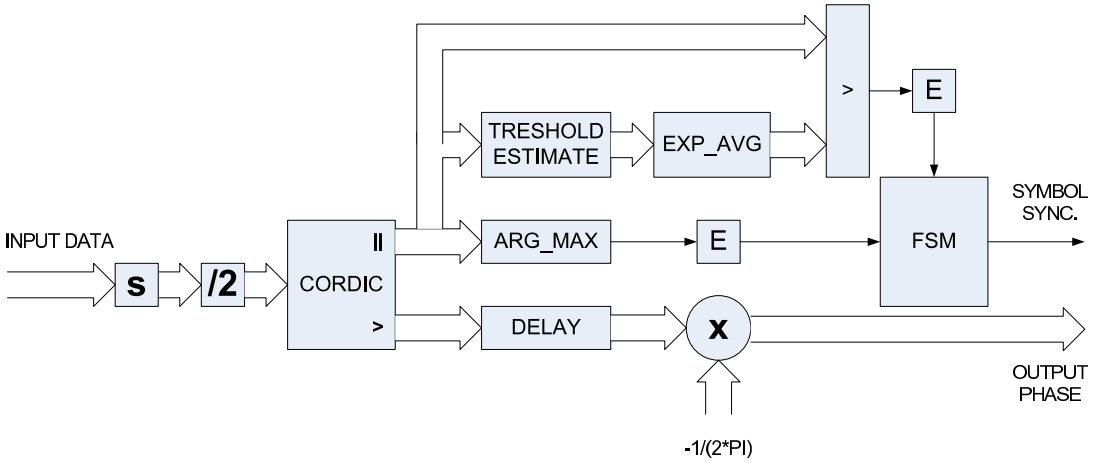

Figure 4: P1 symbol position detector block diagram

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

2.3 P1 Symbol Decoding

When the P1 symbol position is detected the P1 symbol data have to be stored and decoded. The data are stored into dual-port block memory. In the case of P1 symbol the memory is 1024 word length to store all symbol data. One memory port is connected to the delay link as shown in Figure 2 to store symbol data and the second port is available for decoding. The stored data are transformed using FFT operation. Next, bit sequence is generated from transformed data and finally S1 and S2 are estimated from the bit sequence. The P1 symbol decoder is shown in Figure 5.

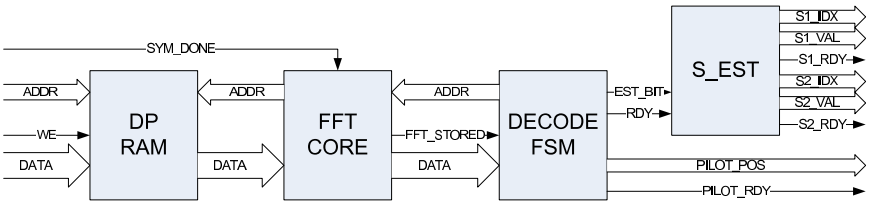

Figure 5: P1 symbol decoder block diagram

Each decoding step is represented and implemented as one component:

FFT Core. The FFT core implements “FFT shift” operation. This operation is done as standard FFT operation but the output data are reordered. Reordering is performed as swapping of low and high half of FFT output data frame.

The Xilinx FFT Core in Radix-4 Burst mode is used for the FFT operation. The input data are indexed and load automatically by the core from input memory when the start signal is activated. This phase is called as load phase. The dual-port memory with P1 symbol data is used as FFT input memory in this case. Data transformation itself starts when all data are loaded and the done signal indicates when the transformation is finished. The output data are unloaded and stored automatically to output memory during next load phase starts. The output data must be stored before the next P1 has been detected so that control logic provides next load phase automatically without any valid input data after data transformation.

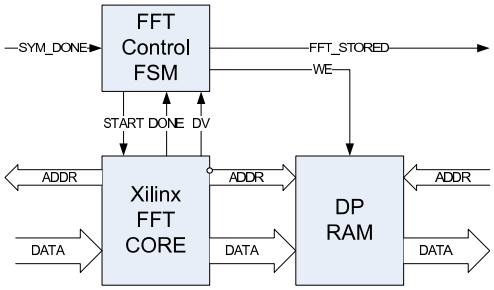

Figure 6: FFT unit of the P1 decoder (block diagram)

The reordering of output data frame is done simply with inverting the most significant bit of output address value to the output memory. The new address value points to the opposite half of memory than before. The dual-ported memory is used to store data. One memory port is connected to the FFT core and the second port provides interface to the next decoding block. The basic block diagram of FFT core is shown in Figure 6.

FSM Decoder. The decoder finite state machine generates decoded bit sequence for estimation of S1 and S2. The basic block diagram of decoder finite state machine module is shown in Figure 8. The bit sequence decoding is based on pilot angle value difference in frequency spectrum. The

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

frequency spectrum is computed by previous FFT core block and pilot positions are stored in ROM. The spectrum can be shifted and that is why the appropriate position of the pilots must be found first. The ROM values must be compensated with appropriate offset. The compensation offset value is used as integer part of frequency offset estimation.

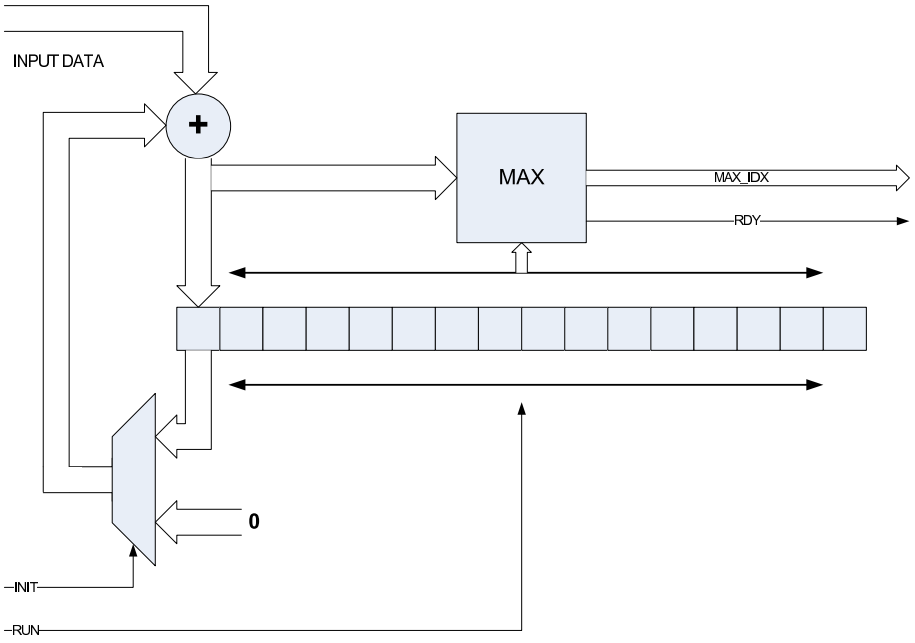

Figure 7: Position correlator of the P1 decoder (block diagram)

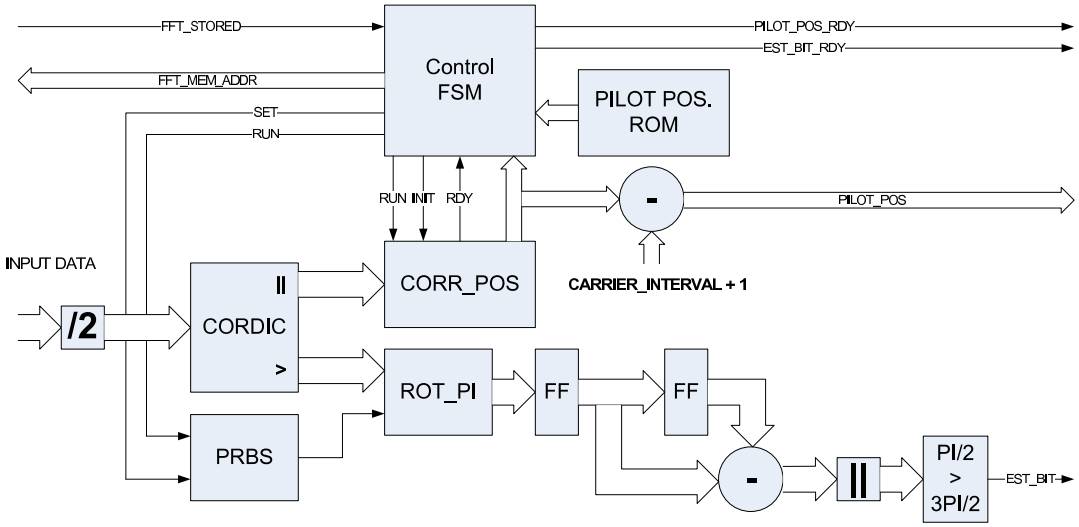

Figure 8: P1 decoder finite state machine (FSM) block diagram

The decoder finite state machine behavioral can be logically divided into two phases—pilot position correlation (see Figure 7) and decoding itself (see Figure 8).

S Estimator. To estimate appropriate S indexes, the input bit sequence is compared with patterns stored in ROM. For S1, there are eight 64-bit length patterns and for S2, sixteen 256-bit length patterns. The most likelihood pattern index is decoded as S1 and S2 index value. The input bit sequence has format shown in Figure 9. The S1 patterns are compared with result of logical OR operation of part S1_low and S1_high and S2 patterns are compared with part S2 as is. The S1_low part must be stored before S1_high comes in shift register. When the S2_high part

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

comes, the comparison can start. For each pattern there are separate counters which count difference between pattern and appropriate bit sequence. The difference between pattern and S bit sequence is determined using logical exclusive or (XOR). The counter and its index with the lowest value for S1 or S2 is selected as output of the estimator. The basic block diagram is shown in Figure 10.

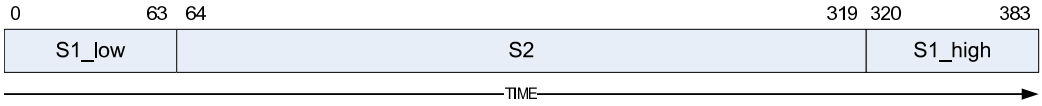

Figure 9: Input data format (bit ordering) of the S-estimator

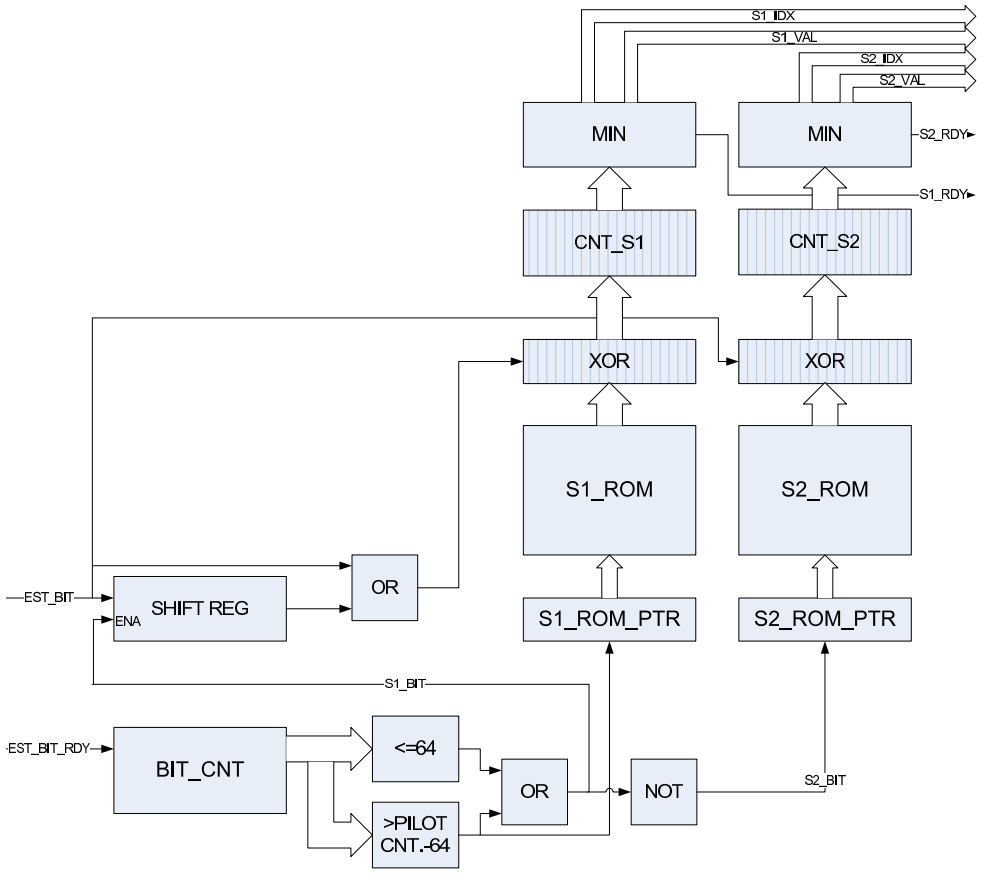

Figure 10: P1 decoder S-estimator block diagram

3 Guard Interval Detector and Time and Frequency Estimator

The guard interval detector determines OFDM symbols guard interval (GI). Guard interval detection is based on computing differences between known FFT size and detected symbol sizes during receiving the T2 frame. The FFT size is decoded from S1 and S2 indexes in P1 processing and symbol size is detected by time and frequency estimator.

The correlator needs to set the FFT size and appropriate symbol GI. The FFT size is decoded from S1 and S2 indexes but the GI is unknown so the smallest possible GI is set as the initial guard interval. When initial GI does not correspond with the actual guard interval, the correlator output peak has

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

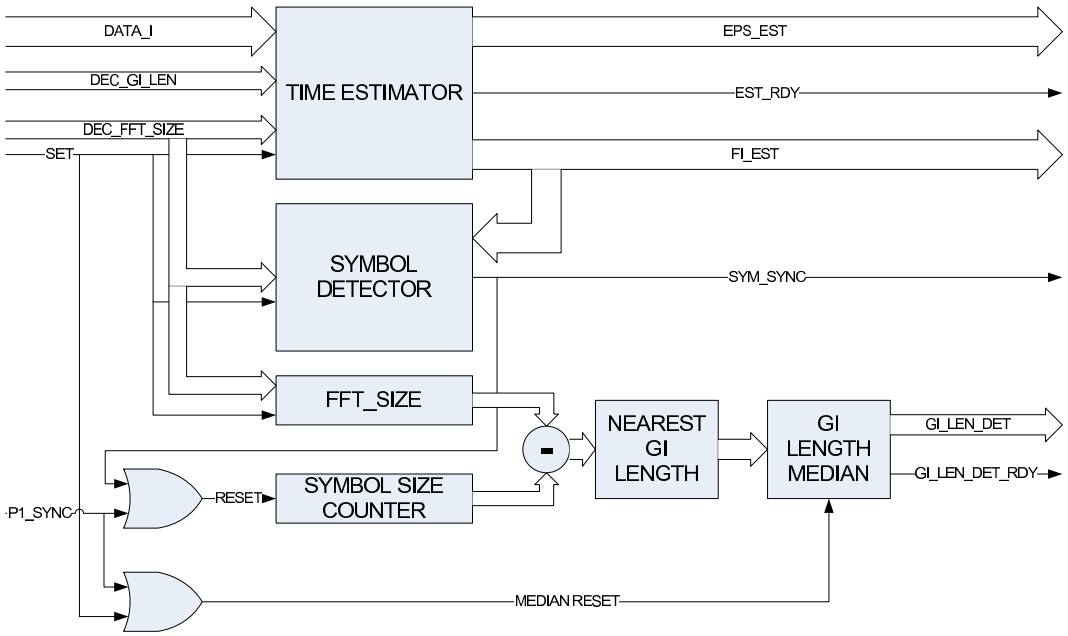

Figure 11: Guard interval detector block diagram

plateau but peak start position is not changed. Correlator output peaks are detected and distance between two last peaks is computed. Difference between the distance and FFT size takes an estimated value of GI. There is only one set of possible GI values and the closest value is selected as the result for actual OFDM symbol. Median of GI values over one T2 frame is considered as the result. Median function is implemented as an array where each position corresponds with one possible GI value. When GI is detected, appropriate value is incremented. Position with maximal value is selected and GI is determined. The array is reset when a new P1 symbol is detected. The GI detector block diagram is shown in Figure 11.

Time and Frequency Estimator. The delay components are implemented as circular buffers with configurable length which allows for running correlation for any FFT size and GI length combinations. Circular buffer lengths (FFT size and GI length) can be set during run time. Correlator contains two running average components whose output must be scaled according to the inner circular buffer length. The inner circular buffer length is equal to actual GI length. Scaling is provided by divisor component. To compute magnitude and phase of complex number the CORDIC component is used. The magnitude output of correlator is used to OFDM symbol detection and phase output is used as fractional part of frequency offset correction. The time and frequency estimator block diagram is shown in Figure 12.

OFDM Symbol Position Detector. OFDM symbol detector works on the same principle as P1 detector described in Section 2.2. The output of time and frequency estimator is used as input of the symbol detector. When FFT size and GI length is properly set the time estimator output peak has no plateau and peak position can be found very precisely. Instead of gradient peak detection method (suitable for P1 correlator peak detection), the method with precise maximum detection is used. The detector block diagram is shown in Figure 13.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

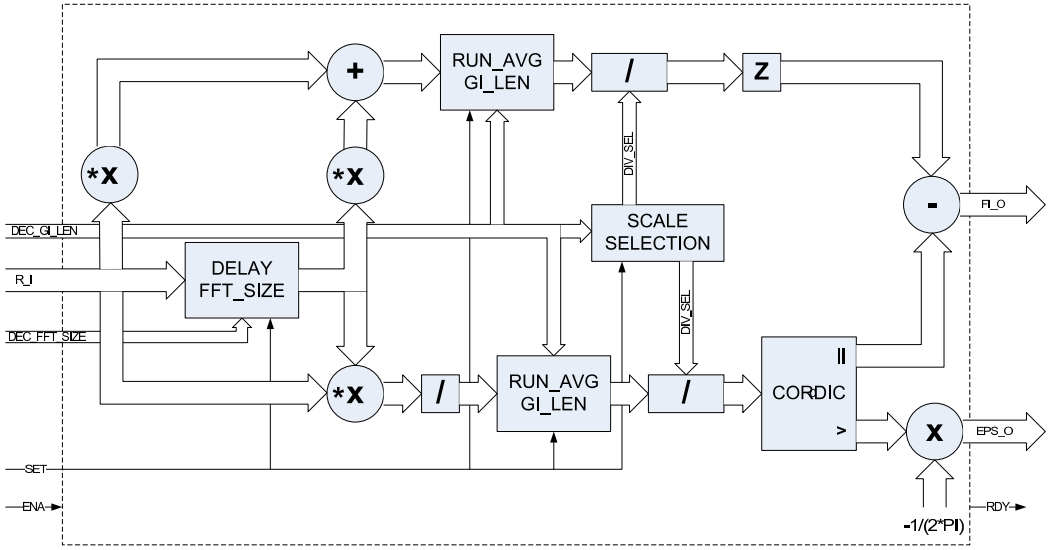

Figure 12: Time and frequency estimator block diagram

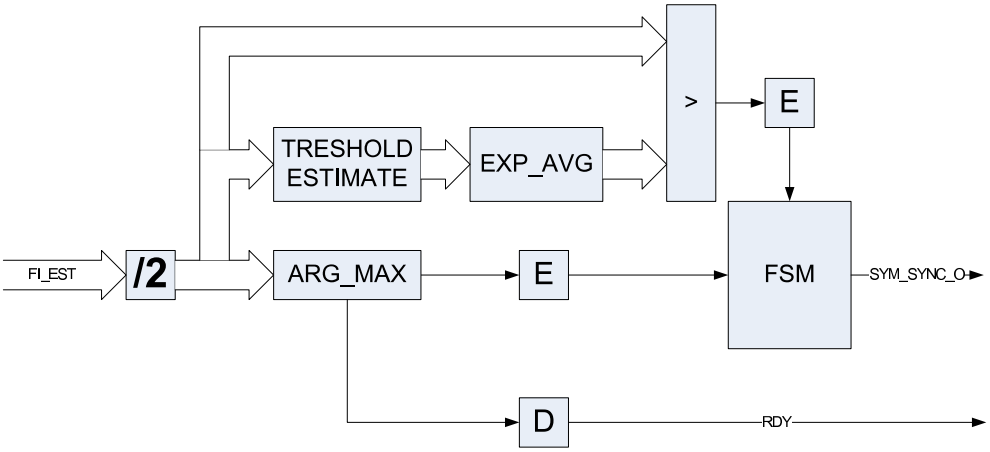

Figure 13: OFDM symbol position detector block diagram

4 OFDM Symbol Count Detector

The OFDM symbol detector determines the count of symbols in the T2 frame. The T2 frame contains P1 symbol at the beginning and several P2/data symbols. The detector contains one counter for counting P1 symbol length, one counter for counting symbol size and one for symbol count. When P1 symbol is detected all counters are reset and then one internal counter counts up to P1 symbol length and the second one periodically up to OFDM symbol length until the next P1 symbol is detected. When the second counter reaches the symbol size value the count of detected symbols is incremented. The symbol size is determined as the FFT size plus GI length. Rounding mechanism is implemented due to lower P1 symbol position detection accuracy and different detection latency of P1 symbol and of OFDM symbol. When difference between symbol size counter and symbol size is less than actual GI length the “round” signal is activated. When the next P1 symbol is detected and “round” signal is activated the symbol count output value should be incremented. The symbol count detector block diagram is shown in Figure 14.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

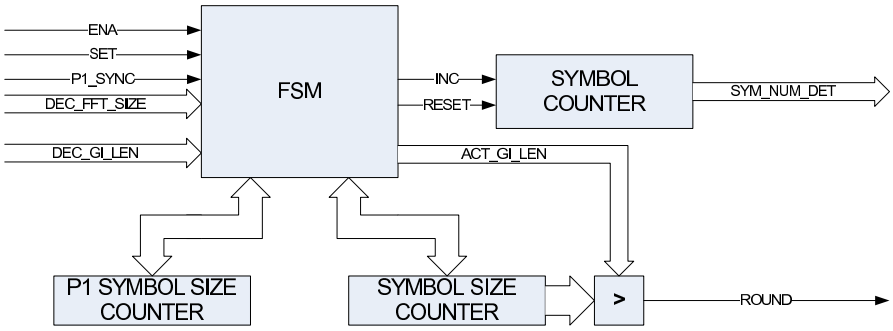

Figure 14: OFDM symbol count detector block diagram

5 Frame Parameters Controller

The frame parameter controller controls frame parameter values (FFT size, GI length, OFDM symbol count) setting and enables frame synchronization to allow decoding data. The symbol count value is properly detected when FFT size and GI length is known, the GI length is properly detected when FFT size is known and FFT size is decoded independently of P1 symbol. The controller internal parameter values are initially set to default values. The initial FFT size is 2K, initial GI length is 64 and symbol count is set to maximal possible value. When any parameter is decoded/detected and has different value than initial value the set signal is activated and new parameter values are set. In case new FFT size value is decoded the GI length and symbol count are set to initial values. The frame parameters controller block diagram is shown in Figure 15.

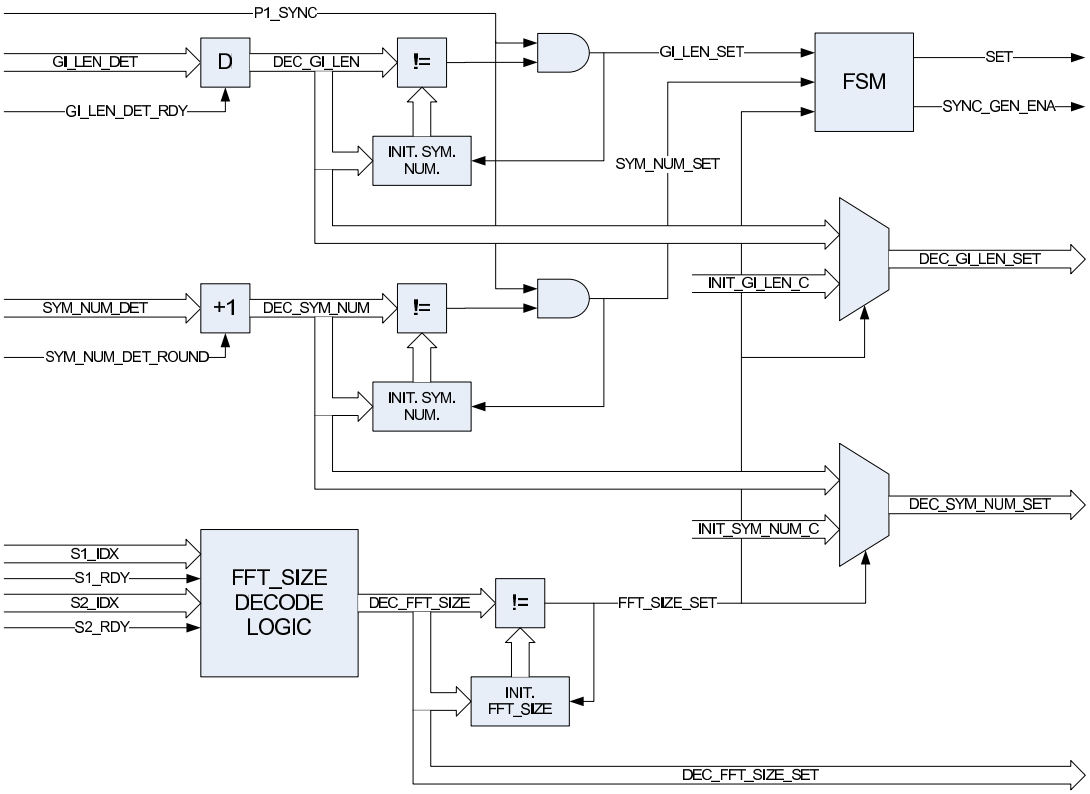

Figure 15: Frame parameters controller block diagram

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

6 Fine Time Synchronization Module

When all T2 frame parameters are known (FFT size, GI length and OFDM symbol count) fine time synchronization can start. Symbol synchronization signals are generated by the finite state automata (FSM) according to the decoded/detected parameters. FSM is started by P1 symbol detection and symbol synchronization signals are generated. These numerically generated signals are compared with real generated signals from P1 symbol detector and OFDM symbol detector. Average time difference of generated and real synchronization signals is used as the FSM correction value. Only the OFDM symbol synchronization signals are used because symbols are detected more precisely than P1 symbol.

Synchronization signal time difference is determined by small FSM with two counters. When generated signal comes earlier than the real signal the counter for generated signal starts counting until real signal comes or counter overcome some tolerance value. Then, the FSM is set into initial state. In the same way the counter for real signal starts counting. The correction is computed as a difference between these two counters when both signals are detected. The maximum detector latency has to be compensated and the latency value is subtracted from the detected difference value. The difference detector block diagram is in Figure 17. The fine time synchronization module block diagram is shown in Figure 16.

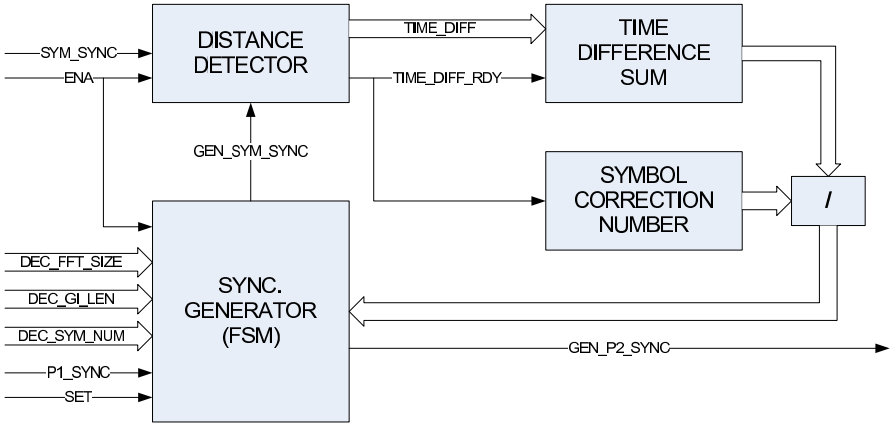

Figure 16: Fine time synchronization module block diagram

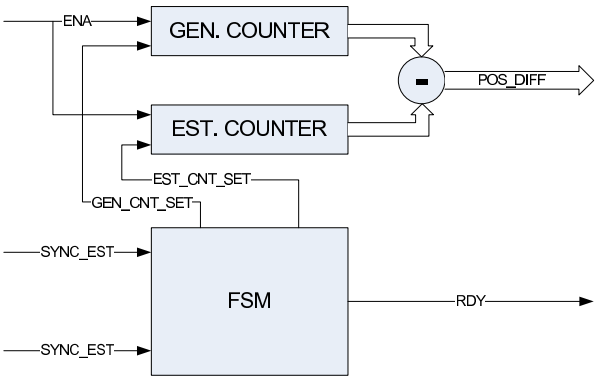

Figure 17: Distance detector block diagram

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

7 P2 and Data Symbol Processing

When first P2 symbol is detected, the PN_SEQ generator is set to initial state and all P2/data symbols in T2 frame are processed. Due to time and frequency estimator and symbol detector latency the input data has to be delayed. The latency is 34 clock cycles. Latency compensation is implemented by cyclic buffer in the top level design. Basic block diagram of P2/data symbols processing is shown in Figure 18.

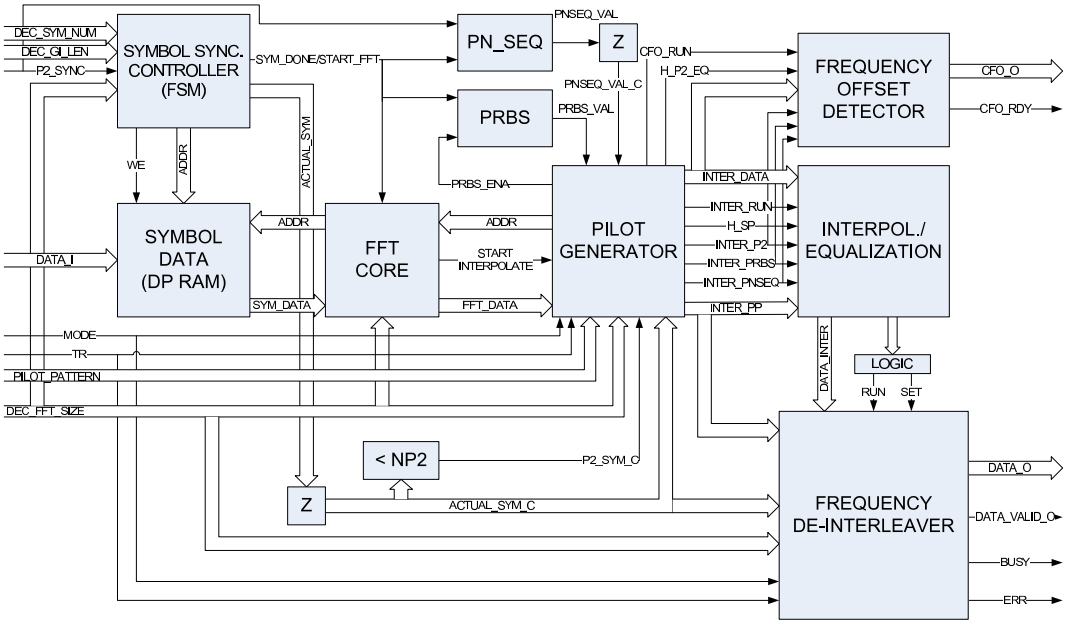

Figure 18: P2 and data symbol processing block diagram

The symbol processing is divided into several functional blocks which provide following operations:

- P2/data symbols time synchronization
- Transformation from time to frequency domain - FFT
- Pilots and reserved tones selection
- Interpolation and equalization
- Carrier frequency offset detection
- Frequency de-interleaving

7.1 Symbol Synchronization Controller

The controller is a simple finite state machine (FSM) which synchronizes symbols in time from the start point represented by first P2/data symbol detection described in Section 6. Each received T2 frame consists of several P2/data symbols and each symbol is preceded by guard interval. FSM drives symbol data storing into FFT core input memory according to actual T2 frame parameters (FFT size, GI length and symbol count). GI length counter, symbol length counter and symbol count counter are implemented in the controller to divide receiving frame in time. The controller enables storing data into the memory and incrementing memory address when symbol data are receiving. Signal “sym_done” is

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

activated when symbol data are stored into the memory. FFT module can start data processing, new PN_SEQ value is generated and PRBS generator is set to the initial state when the signal is activated. Each symbol in the frame has serial number starting with 0. The actual serial number value is passed to the controller output.

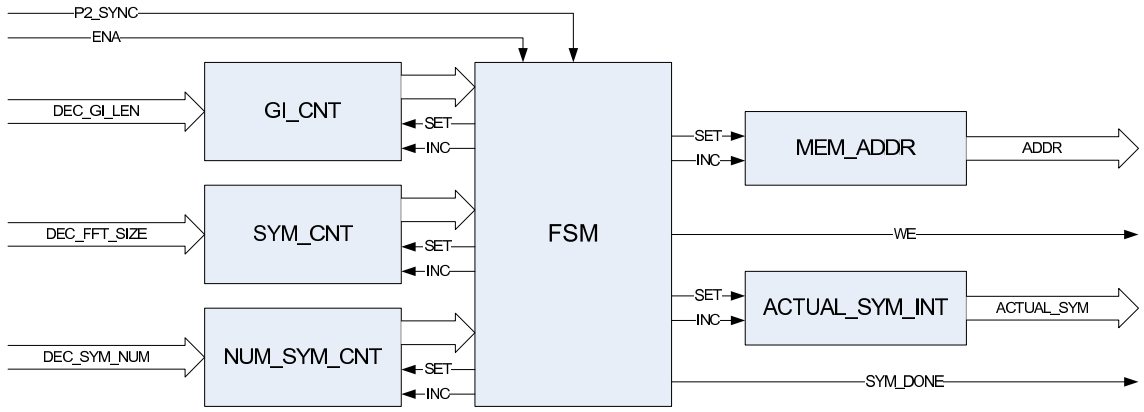

Figure 19: Symbol synchronization controller block diagram

Values of PN_SEQ sequence, P2 symbol indication and symbol serial number are updated before appropriate data are processed, so these values have to be delayed until next symbol is detected. The symbol synchronization controller block diagram is shown in Figure 19.

7.2 FFT Core

The FFT core transforms symbol data from time domain to the frequency domain. The core is implemented in the same way as FFT core in P1 processing described in Section 2.3. The FFT operation is provided with Xilinx FFT Core in Radix-2 Burst mode with no scaling and variable FFT length. FFT core with scaling took bed results during simulations and that is why core with no scaling mode is used. The FFT core block diagram is shown in Figure 20.

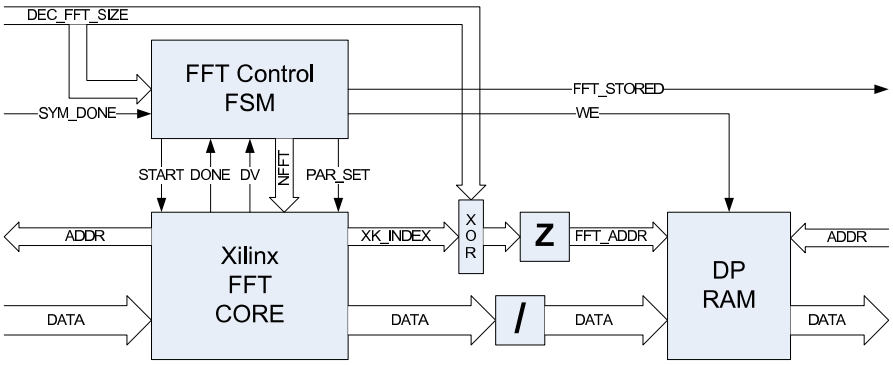

Figure 20: FFT core block diagram

The output data reordering (FFT shift) is done simply with inverting appropriate bit of output address value to the output memory. The actual FFT size has value equal to power of two. The FFT size value directly determines the bit position which value is inverted. The bit inversion is implemented as exclusive or (XOR) between FFT output address value and actual FFT size value shifted by one bit position to the right. FFT address is delayed due to instantiated data scaling components latency.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

7.3 Pilot Generator

Each P2/data symbol has some pilots or reserved tones. When P2/data symbol is transformed into frequency domain, pilots or reserved tones are marked. The pilot generator sequentially reads out the FFT output memory data and signalize whether the data are pilots or reserved tones. Pilots and reserved tones position depends on several parameters such as pilot pattern, FFT size, whether extended mode or tone reservation is enabled, actual symbol serial number and if symbol is P2 symbol. There are two pilot types: scattered and continual. The scattered pilot positions can be determined by an equation and continual pilots are defined by table. Reserved tones are defined by the table. See [1] for details.

Generally, the generator determines FFT memory address space boundaries where data are sequentially read. The first addresses of pilots and reserved tones are determined and next FFT memory data reading is started. If any of pilots or reserved tone address match with the memory address then the pilot/reserved tone address is incremented by next pilot/reserved tone offset. In case of scattered pilot the offset is determined by the logic. Continual pilots and reserved tones offsets are stored in ROMs. Due to ROM latency, there are small FIFOs connected to the memory output which can give data each clock cycle. This implementation prevents that while next offset value is reading from the ROM the memory address belongs to pilot/reserved tone is overcome. Data, pilot/reserved tone flag and demodulation parameters are passed to the interpolator described in Section 7.4.

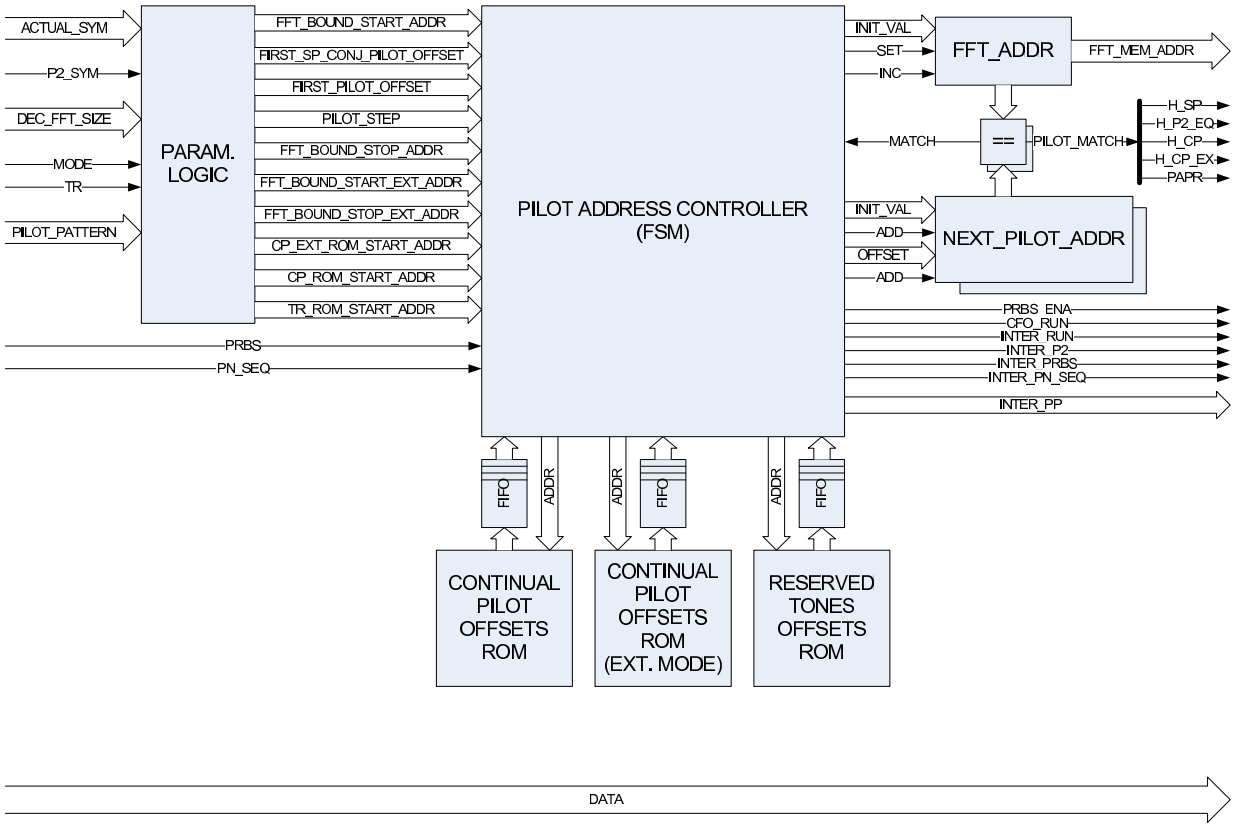

Figure 21: Pilot generator block diagram

Next function of the pilot generator is to drive the PRBS bit sequence generating, drive carrier frequency offset (CFO) detector. In case of CFO detector the FFT memory address space boundaries are resized with carrier frequency interval. This enables to run CFO detector correlation in boundaries of original address space (see the Section 7.5). The FFT core block diagram is shown in Figures 21 and 22.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

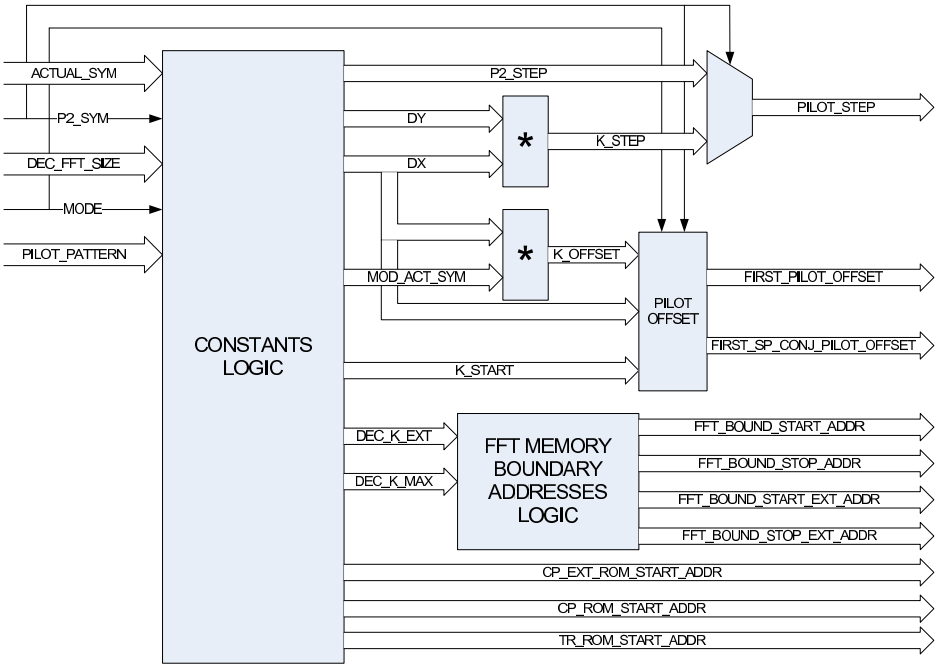

Figure 22: Pilot generator parameters logic block diagram

7.4 Interpolator and Equalizer Module

The interpolator uses linear interpolation method. Interpolation is made between two last scattered pilots. Incoming pilots are demodulated at first. Demodulation depends on pilot pattern, PRBS bit sequence, PN_SEQ bit sequence and whether P2 symbol is interpolated. Transmitted pilot values are constants selected by the demodulation logic. The two last demodulated pilots are held in two registers H1 and H0. Incoming pilot is stored into the register H1 and previous pilot value from H1 is stored into H0 register. Samples between the pilots are interpolated by interpolation function. The interpolator and equalizer module block diagram is shown in Figure 23.

Pilot distance is measured by the logic. If interpolation between two pilots is running and next pilot comes the incoming pilot and measured distance are stored into FIFOs. The whole interpolated data sequence is composed of incoming pilots and interpolated samples. The appropriate pilots and the interpolated samples are selected by multiplexer in time.

Equalization is provided by divider where incoming data are divided with interpolated data sequence. Input data are stored into data FIFO when interpolated data sequence is not ready. The data in the FIFO contain one bit information more. This bit holds information whether data sample is a data cell or not. The data cells are data from a symbol which is not pilot or reserved tone.

7.5 Carrier Frequency Offset Detector

The carrier frequency offset (CFO) detector uses the pilot generator output described in Section 7.3 to detect CFO on P2 symbol. The detector uses auto-correlation based on McNair's algorithm [4].

The metric block based on McNair's algorithm is implemented for each possible CFO in the detector. The block diagram of the metric block is shown in Figure 24. When the pilot generator is reading out FFT output memory data and generates pilot flags, the CFO detector computes metric for each possible offset in the defined interval. The interval is defined by the generic parameter P2_CI. The

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

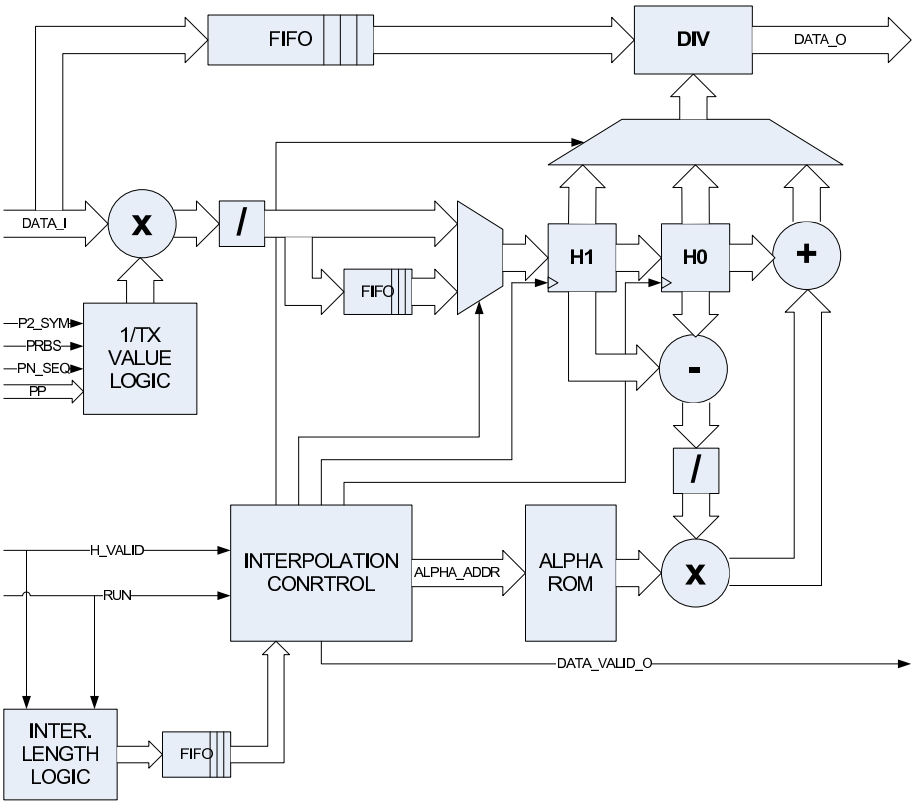

Figure 23: Interpolator and equalizer module block diagram

pilot generator has to read out P2_CI more data samples at the beginning and at the end of its internal address boundaries to read all data for the CFO detector.

The pilot flag has to be shifted in time in positive and negative direction to provide pilot position auto-correlation. Negative shifting means that earlier data are flagged as pilots and positive shifting means that later data are flagged as pilots. In case of negative shifting, the data are delayed and in case positive shifting the flag is delayed. In case of no shifting, the data and flag are directly connected to the appropriate metric block.

The results of each metric block are summarized and metric maximum value is selected at the end of P2 symbol reading. The maximum position takes directly integer CFO value. The carrier frequency offset detector block diagram is shown in Figure 25.

7.6 Frequency De-Interleaver

Frequency de-interleaver reorders data cells in received symbols. Data cell addresses are given by address generator described in [1]. The de-interleaver contains set of address generators for each possible FFT size and appropriate generator is selected according to the actual FFT size.

De-interleaving is made according to the algorithm described in [2]. Additional logic of the address generator generates write and read addresses and read and write enable signals. Data cells are read/write from/into de-interleaver memory according to this algorithm. The address generator with additional logic basic block diagram is shown in Figure 26.

The address generator generates pseudo random sequence of data cell addresses for even (HP0 address) and odd (HP1 address) symbols. The HP0 and HP1 addresses are checked for actual data

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

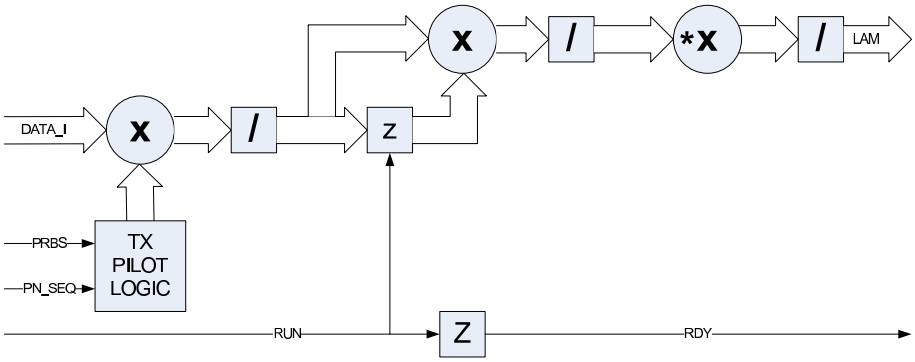

Figure 24: Metric block based on McNair's algorithm block diagram

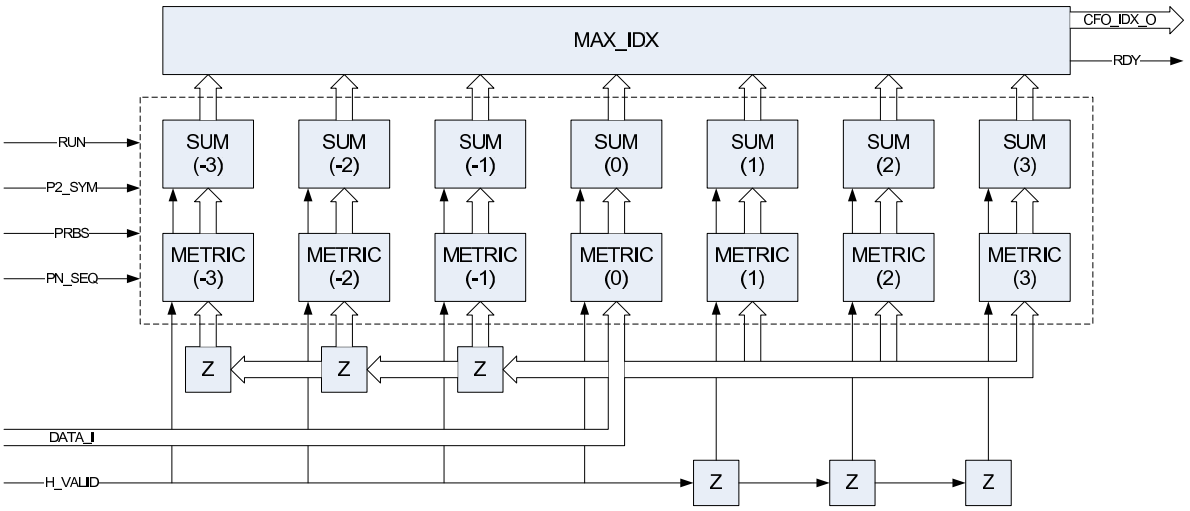

Figure 25: Carrier frequency offset detector block diagram; P2_CI = 3

cell address range C_max. Both addresses have to be valid for incoming data cell. The addresses may not be valid (out of data cell address range) in one address generator iteration step. If one of the addresses is not valid, the second one is stored into one of two FIFOs and the remaining address is generated within the next iteration of address generator.

The address generator is driven by FIFO signals. When FIFOs have free space the generator is running and addresses are pre-fetched into the FIFOs. Signal "fifo_last" indicates last N free positions in the FIFO. The number N of free positions is defined by FIFO generic parameter. Size of the generic parameter is equal to latency of the signal "hp_run" which drives address generation. The size is given by maximal difference between number of valid HP0 and HP1 addresses generated during de-interleaving.

The data cell write/read address is determined from values stored in the FIFOs and from data cell counter (P_CNT). Read enable signal is activated when the determined read address is less or equal to previous data cell address range and write signal is activated when the determined write address is less or equal to actual data cell address range.

The de-interleaver contains one FIFO to store incoming data cells until write address is not ready. When the write address is ready the data cells are stored from the FIFO into the de-interleaver memory. An error is indicated when the FIFO is full. When the read address is ready the data cells from previous symbol are read out from the de-interleaver memory. The de-interleaver block diagram is shown in Figure 23.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

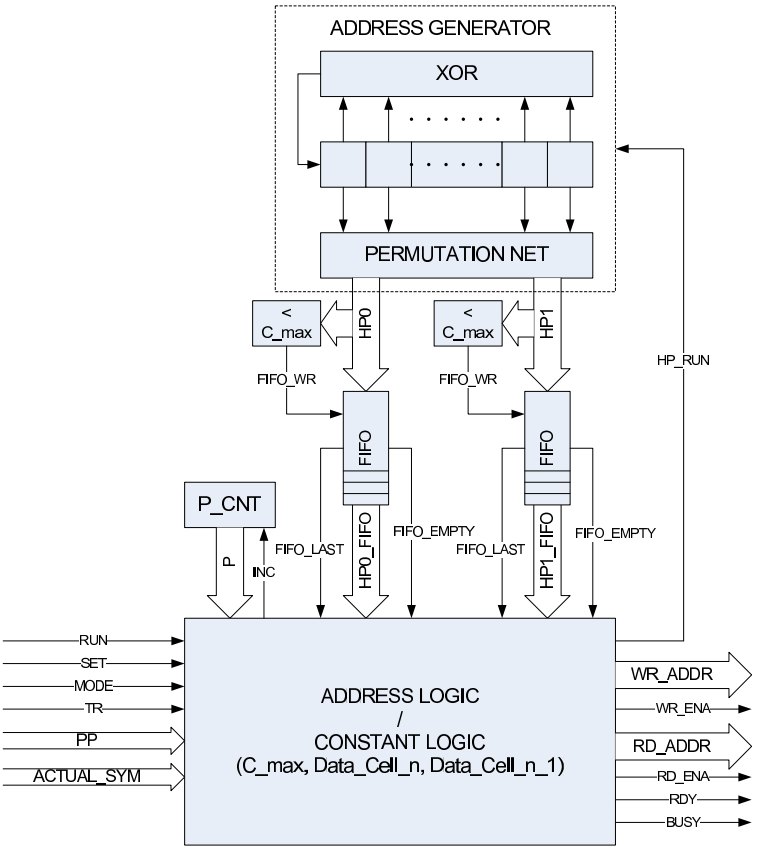

Figure 26: HP generator block diagram

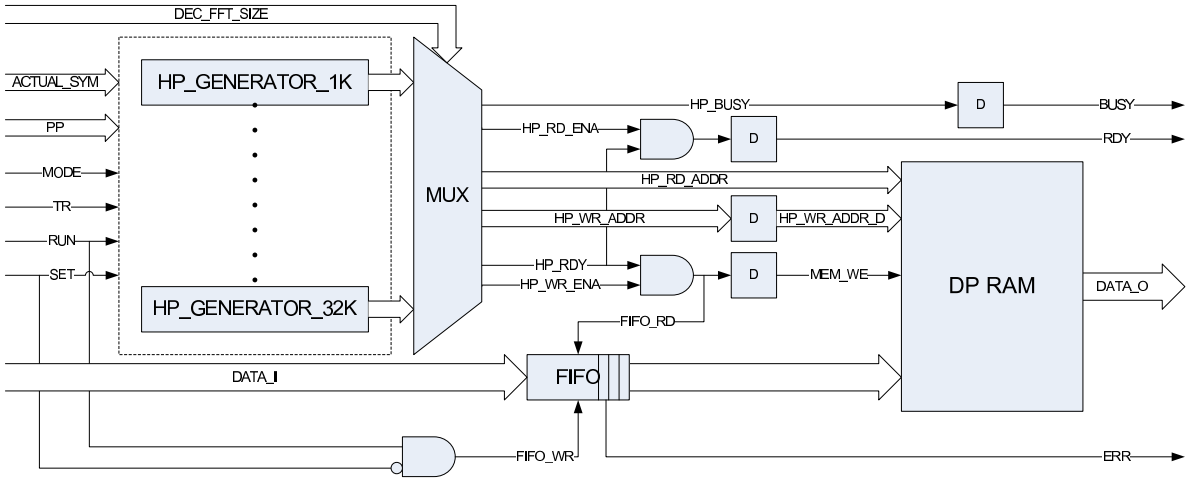

Figure 27: Frequency de-interleaver block diagram

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

References

[1] European Telecommunications Standards Institute, 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France. *ETSI EN 302 755 V1.2.0b: Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)*, draft edition, July 2008. Reference DEN/JTC-DVB-228.

[2] European Telecommunications Standards Institute, 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France. *ETSI TR 102 831 V0.9.6: Digital Video Broadcasting (DVB); Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)*, draft edition, January 2009.

[3] Institute of Information Theory and Automation (UTIA), Pod Vodarenskou vezi 4, 182 08 Prague, Czech Republic. *DVB-T2 Receiver: UTIA Part of the DVB-T2 Compliant Receiver*, June 2009. Technical Manual.

[4] Hanli Zou, B. McNair, and B. Daneshrad. An integrated OFDM receiver for high-speed mobile data communications. In *Global Telecommunications Conference, 2001. GLOBECOM 01. IEEE*, volume 5, pages 3090–3094, 2001.

© 2009 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved