**CHAPTER 1**

**INTRODUCTION**

The DRM Broadcasting system has been developed specifically as a high quality digital replacement for analogue radio broadcasting in the AM and FM/VHF bands, and as such it can be operated with the same channelling and spectrum allocations as urrently employed. The introduction of DRM services allows a broadcaster to provide listeners with significant improvements in service reliability, audio quality and, most importantly, usability. The DRM standard provides many features and facilities which are impossible to replicate in analogue broadcasting. From a technical perspective,a key and revolutionary feature of DRM is the ability to select from a range of transmission modes. This allows the broadcasters to balance or exchange bit-rate capacity, signal robustness, transmission power and coverage. It is possible to do this dynamically, in response to any local changes in the environment, without disturbing the audience.

The frequency bands used for broadcasting below 30 MHz are:

* Low Frequency (LF) band: from 148.5 kHz to 283.5 kHz, in ITU Region 1.
* Medium Frequency (MF) band: from 526.5 kHz to 1606.5 kHz, in ITU Regions 1 and 3 and from 525 kHz to 1705 kHz in ITU Region 2.
* High Frequency (HF) band: a set of individual broadcasting bands in the frequency range 2.3 MHz to 27 MHz, generally available on a Worldwide basis.

These bands offer unique propagation capabilities that permit the achievement of:

* Large coverage areas, whose size and location may be dependent upon the time of day, season of the year or period in the (approximately) 11 year sunspot cycle.
* Portable and mobile reception with relatively little impairment caused by the environment surrounding the receiver.

There is thus a desire to continue broadcasting in these bands, perhaps especially in the case of international broadcasting where the HF bands offer the only reception possibilities which do not also involve the use of local repeater stations. However, broadcasting services in these bands:

* Use analogue techniques.
* Are subject to limited quality.
* Are subject to considerable interference as a result of the long-distance propagation mechanisms which prevail in this part of the frequency spectrum and the large number of users.

As a direct result of the above considerations, there is a desire to affect a transfer to digital transmission and reception techniques in order to provide the increase in quality which is needed to retain listeners who, increasingly, have a wide variety of other programme reception media possibilities, usually already offering higher quality and reliability. In order to meet the need for a digital transmission system suitable for use in all of the bands below 30 MHz, the Digital Radio Mondiale (DRM) consortium was formed in early 1998.In March 2005, the DRM Consortium extended the capability of the DRM system to provide digital radio services at higher transmission frequencies. This range includes:

* 47 MHz to 68 MHz (Band I) allocated to analogue television broadcasting;
* 65.8 MHz to 74 MHz (OIRT FM band);
* 76 MHz to 90 MHz (Japanese FM band);
* 87.5 MHz to 107.9 MHz (Band II) allocated to FM radio broadcasting.

This extension completes the family of digital standards for radio broadcasting.

**1.2 GENERAL CHARACTERISTIC**

**1.2.1 SYSTEM OVERVIEW**

The DRM system is designed to be used at any frequency below 174 MHz, with variable channelization constraints and propagation conditions throughout these bands. In order to satisfy these operating constraints, different transmission modes are available. A transmission mode is defined by transmission parameters classified in two types:

* signal bandwidth related parameters;
* transmission efficiency related parameters.

The first type of parameters defines the total amount of frequency bandwidth for one transmission. Efficiency related parameters allow a trade-off between capacity (useful bit rate) and ruggedness to noise.

**1.2.2 SYSTEM ARCHITECTURE**

This gives a general presentation of the system architecture, based on the synoptic diagram of Fig 1.1 and describes the general flow of different classes of information (audio, data, etc.) and does not differentiate between different services that may be conveyed within one or more classes of information. A detailed description on the distribution of services onto those classes can be found in chapter 3.

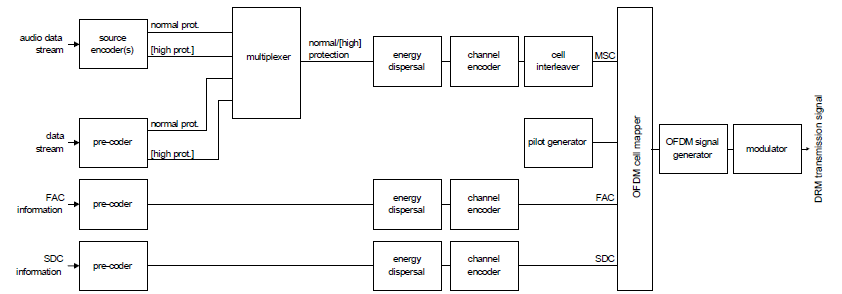


Fig 1.1: Conceptual DRM transmission block diagram

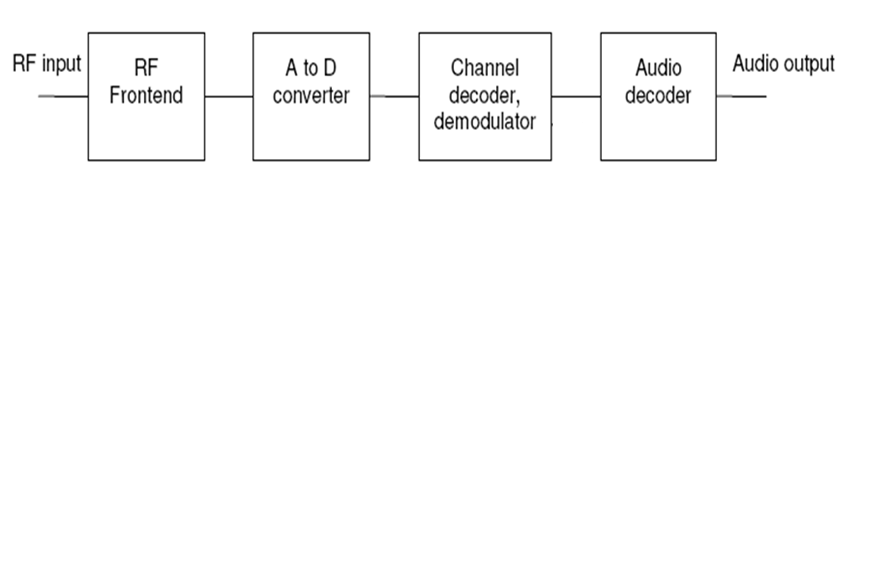


Fig 1.2: DRM Receiver block diagram

The source encoder and pre-coders ensure the adaptation of the input streams onto an appropriate digital transmission format. For the case of audio source encoding, this functionality includes audio compression techniques. The output of the source encoder(s) and the data stream pre-coder may comprise two parts requiring different levels of protection within the subsequent channel encoder. All services have to use the same two levels of protection. The multiplexer combines the protection levels of all data and audio services. The energy dispersal provides a deterministic selective complementing of bits in order to reduce the possibility that systematic patterns result in unwanted regularity in the transmitted signal.

The channel encoder adds redundant information as a means for quasi error-free transmission and defines the mapping of the digital encoded information onto QAM cells Cell interleaving spreads consecutive QAM cells onto a sequence of cells quasi-randomly separated in time and frequency, in order to provide robust transmission in time-frequency dispersive channels. The pilot generator provides means to derive channel state information in the receiver, allowing for a coherent demodulation of the signal. The OFDM cell mapper collects the different classes of cells and places them on the time-frequency grid. The OFDM signal generator transforms each ensemble of cells with same time index to a time domain representation of the signal. Consecutively, the OFDM symbol is obtained from this time domain representation by inserting a guard interval as a cyclic repetition of a portion of the signal. The modulator converts the digital representation of the OFDM signal into the analogue signal in the air. This operation involves digital-to-analogue conversion and filtering that have to comply with spectrum requirements as described in annex E.

**CHAPTER 2**

**TRANSMISSION MODES**

**2.1 SIGNAL BANDWIDTH RELATED PARAMETERS**

The current channel widths for radio broadcasting below 30 MHz are 9 kHz and 10 kHz. The DRM system is designed to be used:

* Within these nominal bandwidths, in order to satisfy the current planning situation;
* Within half these bandwidths (4.5 kHz or 5 kHz) in order to allow for simulcast with analogue AM signals;
* Within twice these bandwidths (18 kHz or 20 kHz) to provide for larger transmission capacity where and when the planning constraints allow for such facility.

The current channel raster (where defined) for radio broadcasting between 30 MHz and 174 MHz is 100 kHz. The DRM system is designed to be used with this raster

**2.2 TRANSMISSION EFFICIENCY RELATED PARAMETERS**

For any value of the signal bandwidth parameter, transmission efficiency related parameters are defined to allow a tradeoff between capacity (useful bit rate) and ruggedness to noise, multipath and Doppler. These parameters are of two types:

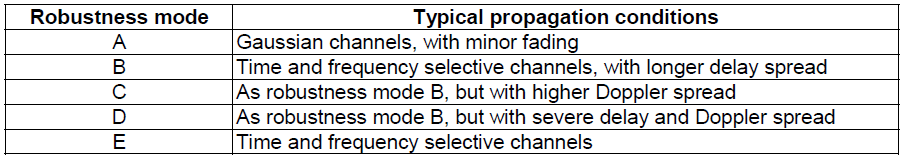
* Coding rate and constellation parameters, defining which code rate and constellations are used to convey data.
* OFDM symbol parameters, defining the structure of the OFDM symbols to be used as a function of the propagation conditions.

**2.3 CODING RATES AND CONSTELLATIONS**

As a function of the desired protection associated within each service or part of a service, the system provides a range of options to achieve one or two levels of protection at a time. Depending on service requirements, these levels of protection may be determined by either the code rate of the channel encoder (e.g. 0.6, etc.), by the constellation order (e.g. 4-QAM, 16-QAM, 64-QAM), or by hierarchical modulation.

**2.4 OFDM PARAMETER SET**

The OFDM parameter set is presented in this paragraph. These values are defined for different propagation-related transmission conditions to provide various robustness modes for the signal.

Table 1.1: Robustness mode uses

In a given bandwidth, the different robustness modes provide different available data rates. Table 1.1 illustrates typical uses of the robustness modes. The transmitted signal comprises a succession of OFDM symbols, each symbol being made of a guard interval followed by the so-called useful part of the symbol. Each symbol is the sum of *K* sine wave portions equally spaced in frequency. Each sine wave portion, called a "cell", is transmitted with given amplitude and phase and corresponds to a carrier position. Each carrier is referenced by the index *k, k* belonging to the interval [ ] *k*min , *k*max ( *k* = 0 corresponds to the reference frequency of the transmitted signal).The time-related OFDM symbol parameters are expressed in multiples of the elementary time period *T* , which is equal to 831/3 μs. These parameters are:

* *Tg* : duration of the guard interval.
* *Ts* : duration of an OFDM symbol.
* *Tu* : duration of the useful (orthogonal) part of an OFDM symbol (i.e. excluding the guard interval).

The OFDM symbols are grouped to form transmission frames of duration*Tf* .

A certain number of cells in each OFDM symbol are transmitted with a predetermined amplitude and phase, in order to be used as references in the demodulation process.

They are called "reference pilots" and represent a certain proportion of the total number of cells.



Table 1.2: OFDM symbol parameters

**CHAPTER 3**

**CHANNEL CODING**

The DRM system consists of three different channels, the MSC, SDC and FAC. Because of the different needs of these channels different coding and mapping schemes shall be applied. An overview of the encoding process is shown in Figure 2.1. The coding is based on a multilevel coding scheme. Due to different error protection needs within one service or for different services within one multiplex different mapping schemes and combinations of code rates are applicable: Unequal Error Protection (UEP) and Equal Error Protection (EEP) are available and can be combined with hierarchical modulation. Equal error protection uses a single code rate to protect all the data in a channel. EEP is mandatory for the FAC and SDC. Instead of EEP, unequal error protection can be used with two code rates to allow the data in the Main Service Channel to be assigned to the higher protected part and the lower protected part. When using hierarchical modulation three mapping strategies are applicable to the MSC: the Standard Mapping (SM), the symmetrical Hierarchical Mapping (HMsym) and a mixture of the previous two mappings (HMmix) that results in the real component of the constellation following a Hierarchical Mapping and the imaginary part following a standard one.

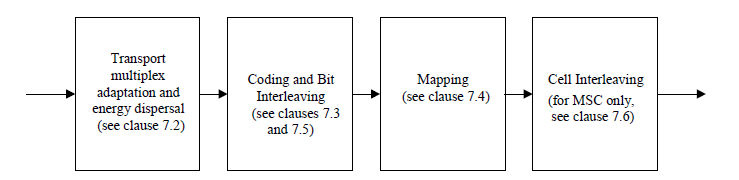
****

Figure 2.1: Functional block diagram of the coding and interleaving

The Hierarchical Mappings split the decodable data stream into two parts: a Very Strongly Protected Part (VSPP) and a Standard Protected Part (SPP). The SM method only consists of a SPP. In any case, up to two different overall code rates shall be applied to the SPP of the MSC. For the FAC and SDC only SM is allowed.

**3.1 MULTIPLEX DEFINITION**

**3.1.1 INTRODUCTION**

The DRM transmission super frame consists of three channels: the Main Service Channel (MSC), the Fast Access Channel (FAC), and the Service Description Channel (SDC). The MSC contains the data for the services. The FAC provides information on the channel width and other such parameters and also provides service selection information to allow for fast scanning. The SDC gives information on how to decode the MSC, how to find alternative sources of the same data, and gives the attributes of the services within the multiplex. It can include links to analogue simulcast services.

**3.1.2 MAIN SERVICE CHANNEL (MSC)**

**3.1.2.1 INTRODUCTION**

The Main Service Channel (MSC) contains the data for all the services contained in the DRM multiplex. The multiplex may contain between one and four services, and each service may be either audio or data. The gross bit rate of the MSC is dependent upon the DRM channel bandwidth and the transmission mode.

**3.1.2.2 STRUCTURE**

The MSC contains between one and four streams. Each stream is divided into logical frames. Audio streams comprise compressed audio and optionally they can carry text messages. Data streams may be composed of data packets, carrying information for up to four "sub-streams". An audio service comprises one audio stream and optionally one data stream or one data sub-stream. A data service comprises one data stream or one data sub-stream. Each logical frame generally consists of two parts, each with its own protection level. The lengths of the two parts are independently assigned. Unequal error protection for a stream is provided by setting different protection levels to the two parts. For robustness modes A, B, C and D, the logical frames are each 400 ms long. If the stream carries audio, the logical frame carries the data for one audio super frame. For robustness mode E, the logical frames are each 100 ms long. If the stream carries audio, the logical frame carries the data for either the first or the second part of one audio super frame containing the audio information for 200 ms duration. Since, in general, the stream may be assigned two protection levels, the logical frames carry precisely half of the bytes from each protection level. The logical frames from all the streams are mapped together to form multiplex frames of the same duration, which are passed to the channel coder. In some cases, the first stream may be carried in logical frames mapped to hierarchical frames. The multiplex configuration is signalled using the SDC. Annex M contains some examples of different MSC configurations.

**3.1.2.3 BUILDING THE MSC**

The MSC consists of a sequence of multiplex frames, and if hierarchical modulation is in use a sequence of hierarchical frames also. The multiplex frames and hierarchical frames are passed separately to the channel coder.

**3.1.2.4 MULTIPLEX FRAMES**

The multiplex frames are built by placing the logical frames from each non-hierarchical stream together. The logical frames consist, in general, of two parts each with a separate protection level. The multiplex frame is constructed by taking the data from the higher protected part of the logical frame from the lowest numbered stream (stream 0 when hierarchical modulation is not used, or stream 1 when hierarchical modulation is used) and placing it at the start of the multiplex frame. Next the data from the higher protected part of the logical frame from the next lowest numbered stream is appended and so on until all streams have been transferred. The data from the lower protected part of the logical frame from the lowest numbered stream (stream 0 when hierarchical modulation is not used, or stream 1 when hierarchical modulation is used) is then appended, followed by the data from the lower protected part of the logical frame from the next lowest numbered stream, and so on until all streams have been transferred. The higher protected part is designated part A and the lower protected part is designated part B in the multiplex description. The capacity of the multiplex frame is larger than or equal to the sum of the logical frames from which it is formed. The remainder, if any, of the multiplex frame shall be filled with 0s. These bits shall be ignored by the receiver.

NOTE: No padding bits are inserted between the end of part A and the beginning of part B. The capacity of part A of the multiplex frame is equal to the sum of the higher protected parts of the logical frames, but as a result of restrictions introduced by the channel encoding procedure applied for DRM (see chapter 3.5.1), some of the bits nominally belonging to the lower protected part B of a multiplex frame might in fact be protected at the higher level.

**3.1.2.5 HIERARCHICAL FRAMES**

The hierarchical frames only exist when hierarchical modulation is used. They are built by taking the data from the logical frame from stream 0 and placing it at the start of the hierarchical frame. The capacity of the hierarchical frame is larger than or equal to the logical frame from which it is formed. The remainder, if any, of the hierarchical frame shall be filled with 0s. These bits shall be ignored by the receiver.

**3.1.2.6 RECONFIGURATION**

The multiplex may be reconfigured at transmission super frame boundaries. A reconfiguration of the multiplex occurs when the channel parameters in the FAC are changed, or when the services in the multiplex are reorganized. The new configuration is signalled ahead of time in the SDC and the timing is indicated by the reconfiguration index in the FAC.

**3.2 FAST ACCESS CHANNEL (FAC)**

**3.2.1 INTRODUCTION**

The FAC is used to provide information on the channel parameters required for the demodulation of the multiplex as well as basic service selection information for fast scanning. The channel parameters (for example the spectrum occupancy and interleaving depth) allow a receiver to begin to decode the multiplex effectively. It also contains information about the services in the multiplex to allow the receiver to either decode this multiplex or change frequency and search again.

**3.2.2 STRUCTURE**

Each transmission frame contains an FAC block. An FAC block contains parameters that describe the channel and parameters to describe either one or two services along with a CRC. For robustness modes A, B, C and D, one set of service parameters shall be transmitted and for robustness mode E, two sets of service parameters shall be transmitted. When more services are carried in the multiplex than can be described within one FAC block, a number of FAC blocks are required to describe all the services, see chapter 3.5

**3.2.3 CHANNEL PARAMETERS**

The channel parameters are as follows:

* Base/Enhancement flag 1 bit
* Identity 2 bits
* RM flag 1 bit
* Spectrum occupancy 3 bits
* Interleaver depth flag 1 bit
* MSC mode 2 bits
* SDC mode 1 bit
* Number of services 4 bits
* Reconfiguration index 3 bits
* Toggle flag 1 bit
* rfu 1 bit

The following definitions apply:

* Base/Enhancement flag :this 1-bit flag indicates whether the transmission is the base or enhancement layer as follows:
* 0: Base layer - decodable by all DRM receivers.
* 1: Enhancement layer - only decodable by receivers with enhancement layer capabilities.
* Identity: this 2-bit field identifies the current FAC block within the transmission super frame and also validates the SDC AFS index as follows:
* 00: first FAC block of the transmission super frame and SDC AFS index is valid.
* 01: intermediate FAC block of the transmission super frame.
* 10: last FAC block of the transmission super frame.
* 11: first FAC block of the transmission super frame and SDC AFS index is invalid.

NOTE: Either one or two intermediate FAC blocks may be present within a transmission super frame depending on the value of the RM flag. Intermediate FAC blocks can be distinguished by using the Toggle flag.

* RM flag: this 1-bit field indicates the robustness mode as follows:
* 0: Robustness modes A, B, C or D; FAC block contains one set of service parameters.
* 1: Robustness mode E; FAC block contains two sets of service parameters.

The interpretation of the Spectrum occupancy, Interleaver depth flag, MSC mode, SDC mode and Toggle flag parameters are dependent on the value of the RM flag as detailed in the following clauses.

* Spectrum occupancy**:** this 3-bit field, coded as an unsigned integer, specifies the nominal channel bandwidth andconfiguration of the digital signal as follows. See also clause 8.
* RM flag = 0:
* 0: 4.5 kHz.
* 1: 5 kHz.
* 2: 9 kHz.
* 3: 10 kHz.
* 4: 18 kHz.
* 5: 20 kHz.
* other values reserved.
* RM flag = 1:
* 0 : 100 kHz.
* other values reserved.
* Interleaver depth flag: this 1-bit flag indicates the depth of the time interleaving as follows:
* RM flag = 0:
* 0: 2 s (long interleaving).
* 1: 400 ms (short interleaving).
* RM flag = 1:
* 0: 600 ms.
* 1: reserved.
* MSC mode: this 2-bit field indicates the modulation mode in use for the MSC as follows:
* RM flag = 0:
* 00: 64-QAM, no hierarchical.
* 01: 64-QAM, hierarchical on I.
* 10: 64-QAM, hierarchical on I&Q.
* 11: 16-QAM, no hierarchical.
* RM flag = 1:
* 00: 16-QAM, no hierarchical.
* 01: reserved.
* 10: reserved.
* 11: 4-QAM, no hierarchical.
* SDC mode**:** this 1-bit field indicates the modulation mode and code rate in use for the SDC as follows:
* RM flag = 0:
* 0: 16-QAM, code rate = 0.5.
* 1: 4-QAM, code rate = 0.5.
* RM flag = 1:
* 0: 4-QAM, code rate = 0.5.
* 1: 4-QAM, code rate = 0.25.
* Number of services: this 4-bit field indicates the number of audio and data services as follows:
* 0000: 4 audio services.
* 0001: 1 data service.
* 0010: 2 data services.
* 0011: 3 data services.
* 0100: 1 audio service.
* 0101: 1 audio service and 1 data service.
* 0110: 1 audio service and 2 data services.
* 0111: 1 audio service and 3 data services.
* 1000: 2 audio services.
* 1001: 2 audio services and 1 data service.
* 1010: 2 audio services and 2 data services.
* 1011: reserved.
* 1100: 3 audio services.
* 1101: 3 audio services and 1 data service.
* 1110: reserved.
* 1111: 4 data services.
* Reconfiguration index: this 3-bit field indicates the status and timing of a multiplex reconfiguration. A non-zero value indicates the number of transmission super frames of the old configuration that are transmitted before the new configuration takes effect.
* Toggle flag: this 1-bit flag shall be used to indicate that this transmission frame may contain the start of an audio superframe as follows:
  + RM = 0:

Toggle flag is fixed to zero

* + RM = 1:

The toggle flag is set to zero for the first and third FAC block of the transmission super frame and to one for the second and fourth FAC block. It may be used in combination with the Identity parameter to distinguish the received transmission frames.

* rfu: this 1 bit flag is reserved for future use of the whole FAC parameter definitions and shall be set to zero until defined.

**3.2.4 SERVICE PARAMETERS**

The service parameters are as follows:

* Service identifier 24 bits.
* Short Id 2 bits.
* Audio CA indication 1 bit.
* Language 4 bits.
* Audio/Data flag 1 bit.
* Service descriptor 5 bits.
* Data CA indication 1 bit.
* rfa 6 bits.

The following definitions apply:

* Service identifier: this 24-bit field indicates the unique identifier for this service.
* Short Id:this 2-bit field indicates the short identifier assigned to this service and used as a reference in the SDC. The Short Id is assigned for the duration of the service and is maintained through multiplex reconfigurations.
* Audio CA indication:this 1-bit flag indicates whether the service uses conditional access as follows:
* 0: No CA system is used for the audio stream (or the service has no audio stream).
* 1: CA system is used for the audio stream.

NOTE 1: The details are provided by the SDC data entity type 2. Every DRM receiver shall check the "Audio CA indication" bit before presenting the audio stream of the audio service.

A non-CA capable DRM receiver shall not try to decode the audio stream if the "Audio CA indication" is set to 1.

* Language:this 4-bit field indicates the language of the target audience as defined in table 53.

NOTE 2: Further languages are also indicated by SDC data entity type 12.

* Audio/Data flag:this 1-bit flag indicates whether the service is audio or data as follows:
  + 0: Audio service.
  + 1: Data service.
* Service descriptor:this 5-bit field depends upon the value of the Audio/Data flag as follows:
  + 0: Programme type.
  + 1: Application identifier.

Regardless of the value of the Audio/Data flag, the value 31 (all bits set to 1) indicates that a standard DRM receiver should skip this broadcast and continue to scan for services.

NOTE 3: This is to allow for engineering test transmissions to be ignored by standard receivers.

* Programme type: this 5-bit field indicates the programme type of an audio service.
* Application identifier:this 5-bit field indicates the application identifier of a data service.
* Data CA indication:this 1-bit flag indicates whether the service uses conditional access as follows:
  + 0: No CA system is used for the data stream/sub-stream (or the service has no data stream/sub-stream).
  + 1: CA system is used for the data stream/sub-stream.

NOTE 4: The details are provided by the SDC data entity type 2.

Every DRM receiver shall check the "Data CA indication" bit before presenting the data stream/sub-stream of the audio or data service. A non-CA capable DRM receiver shall not try to decode the data stream/sub-stream if the "Data CA indication" is set to 1.

* rfa:these 6 bits are reserved for future additions and shall be set to zero until defined.

**3.3 CRC**

The 8-bit Cyclic Redundancy Check shall be calculated on the channel and service parameters. It shall use the generator polynomial *G8(x) = x8 + x4 + x3 + x2 + 1.* When the RM flag = 0, the CRC is calculated over 64-bits formed by concatenating the 20-bits of channel parameters and the 44-bits of service parameters. When the RM flag = 1, the CRC is calculated over 112-bits formed by concatenating the 20-bits of channel parameters, the 88-bits of service parameters (2 sets of 44-bits) and 4-bits set to zero. These 4-bits are used to calculate the CRC but are not forwarded for coding and transmission.

**3.4 FAC REPETITION**

The FAC channel parameters shall be sent in each FAC block. The FAC service parameters for one or two services shall be sent in each FAC block. When more than one FAC block is needed to signal all the services in the multiplex, the repetition pattern is significant to the receiver scan time. When all services are of the same type (e.g. all audio or all data) then the services shall be signalled sequentially. In the case when there is only one service and the FAC block signals two sets of service parameters, both sets shall contain identical content.

**3.5 TRANSPORT MULTIPLEX ADAPTATION**

The different channels (MSC, SDC, FAC) are processed in the channel coding independently. The vector length *L* for processing equals one FAC block for the FAC,one SDC block for the SDC or one multiplex frame for the MSC.

**3.5.1 MSC**

The number of bits *LMUX* per multiplex frame is dependent on the robustness mode, spectrum occupancy and constellation:

* when using one protection level (EEP) it is given by:

*LMUX* = *L*2

* when using two protection levels (UEP) it is given by:

*LMUX* = *L*1 + *L*2

where the number of bits of the higher protected part is *L1* and the number of bits of the lower protected part is *L2*.

* when using HMsym or HMmix the number of very strongly protected bits is given by *LVSPP*.

*L1*, *L2* and *LVSPP* are calculated as follows:

* SM:
  + Pmax is the number of levels (4-QAM: Pmax = 1; 16-QAM: Pmax = 2; 64-QAM: Pmax = 3).
  + RXp is the numerator of the code rate of each individual level.
  + RYp is the denominator of the code rate of each individual level.
  + Rp is the code rate of each individual level.



Table 3.1: Code rate combinations for the MSC with 16-QAM

(robustness modes A, B, C and D)

* HMsym:

* + Pmax =3 is the number of levels for 64-QAM using HMsym.
  + NOTE: A hierarchical mapping scheme can only be used in a 64-QAM signal constellation.
  + RXp is the numerator of the code rate of each individual level.
  + RYp is the denominator of the code rate of each individual level.
  + Rp is the code rate of each individual level.
* HMmix:
  + Pmax = 3 is the number of levels for 64-QAM using HMmix.
  + ,are the numerators of the code rates of each individual level for the real and imaginary component respectively.
  + ,are the denominators of the code rates of each individual level for the real and imaginary component respectively.
  + ,are the code rates of each individual level for the real and imaginary component respectively.
  + ⎣ ⎦ means round towards minus infinity.

The total number *NMUX* of MSC OFDM cells per multiplex frame.

The total number *NMUX* of MSC OFDM cells per multiplex frame when using one protection level (EEP) equals *N2*.

The total number *NMUX* of MSC OFDM cells per multiplex frame when using two protection levels (UEP) equals the addition of the cells of the higher protected part and the lower protected part:

*NMUX* = *N*1 + *N*2

*N1* is the number of OFDM cells used for the higher protected part.

*N2* is the number of OFDM cells used for the lower protected part including the tailbits.

To calculate the number *N1* of OFDM cells in the higher protected part (part A) the following formulae apply:

* SM:
* HMsym:
* HMmix:

where:

* *X* is the number of bytes in part A.
* *RYlcm* is taken from tables 3.1 and 3.2 for SM; from tables 3.3 and 3.4 for HMsym; and from tables 3.2, 3.4 and 3.5 for HMmix.
* means round towards plus infinity.

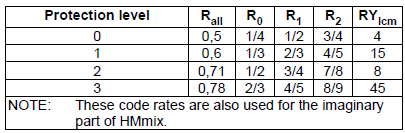


Table 3.2: Code rate combinations for the MSC with 64-QAM

(robustness modes A, B, C and D)

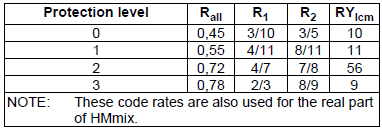


Table 3.3: Code rate combinations for the SPP of MSC with HMsym 64-QAM

(robustness modes A, B, C and D)

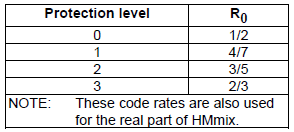


Table 3.4: Code rate combinations for the VSPP of MSC with HMsym 64-QAM

(robustness modes A, B, C and D)

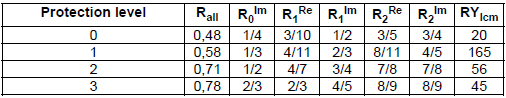


Table 3.5: Code rate combinations for the SPP of MSC with HMmix 64-QAM

(robustness modes A, B, C and D)

To calculate the number *N2* of OFDM cells in the lower protected part (part B) the following formula applies:

*N*2 = *NMUX* − *N*1

The following restrictions shall be taken into account:

*N*1 ∈{0,………. *NMUX* − 20}

*N*2 ∈ {20,…….. *NMUX* }

**3.5.2 FAC**

The number of bits *LFAC* per FAC block equals 72 bits in robustness modes A, B, C and D and 116 bits in robustness mode E. The total number *NFAC* of FAC OFDM cells per FAC block equals 65 in robustness modes A, B, C and D and 244 in robustness mode E.

**3.5.3 SDC**

The number of bits *LSDC* per SDC block is dependent on the robustness mode, spectrum occupancy and constellation. The total number *NSDC* of SDC OFDM cells per SDC block are given in table 3.6. The formulas given in chapter 3.5.1 for the MSC are valid also for the SDC under the constraint of EEP and SM (only 4-QAM: Pmax = 1, 16-QAM: Pmax = 2), i.e. *L*SDC = *L*2 and *N*SDC = *N*2.

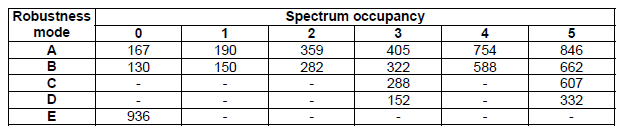
****

Table 3.6: Number of QAM cells *NSDC* for SDC

**3.6 ENERGY DISPERSAL**

The purpose of the energy dispersal is to avoid the transmission of signal patterns which might result in an unwanted regularity in the transmitted signal. For the SDC and FAC, the output of the energy dispersal shall form the input stream ui to the corresponding multilevel coding process. The output of the energy dispersal acting on the MSC multiplex frame shall form the standard protected input stream ui to the multilevel coding process for the MSC. The output of the energy dispersal acting on the hierarchical frame (if present) shall form the very strongly protected input stream u'i to the same multilevel coding process. Energy dispersal shall be applied on the different channels (MSC, SDC, FAC) in order to reduce the possibility that systematic patterns result in unwanted regularity in either the transmitted signal or in any digital processing, this by providing a deterministic selective complementing of bits. The individual inputs of the energy dispersal scramblers shown in figure 3.1 shall be scrambled by a modulo-2 addition with a Pseudo-Random Binary Sequence (PRBS), prior to channel encoding. The PRBS is defined as the output of the feedback shift register of figure 3.1. It shall use a polynomial of degree 9, defined by:

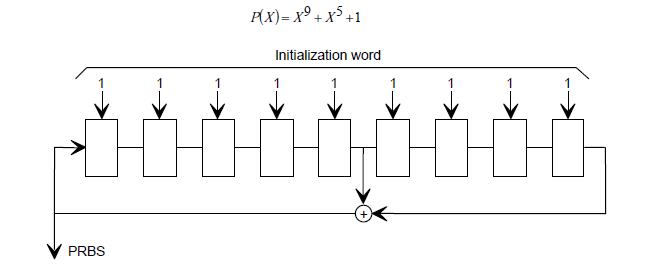
**

Figure 3.1: PRBS generator

The initialization word shall be applied in such a way that the first bit of the PRBS is obtained when the outputs of all shift register stages are set to value "1"; the first 16 bits of the PRBS are given in table 3.7.

****

Table 3.7: First 16 bits of the PRBS

The FAC, SDC and MSC shall be processed by the energy dispersal scramblers as follows:

* The vector length for processing equals one FAC block for the FAC, one SDC block for the SDC and one multiplex frame and one hierarchical frame for the MSC.
* The block length of the FAC is dependent on the robustness mode; the block lengths for the SDC and MSC are dependent on the robustness mode, spectrum occupancy and constellation, see chapter 3.5.
* The four blocks shall be processed independently. The input vector shall be scrambled with the PRBS, the first bit of the vector being added modulo 2 to the PRBS bit of index 0.

The scramblers of the different channels are reset as follows:

* FAC: every FAC block;
* SDC: every SDC block;
* MSC: every multiplex frame for the standard protected part, every hierarchical frame for the very strongly protected part.

**CHAPTER 4**

**TAG ITEMS SPECIFYING DRM MULTIPLEX**

**4.1 ROBUSTNESS MODE (robm)**

This TAG item, as shown in figure 4.1, shall be included in every RSCI TAG packet and is therefore mandatory for all RX\_STAT profiles. If the receiver is not in synchronization an empty TAG item shall be transmitted.

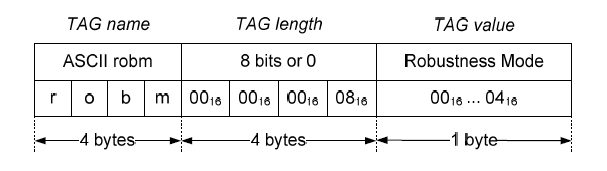


Figure 4.1: Robustness mode

Robustness mode:the current robustness mode as detected by the decoding algorithms.

The TAG value shall be encoded as given in table 4.1. All other values are reserved for future use.

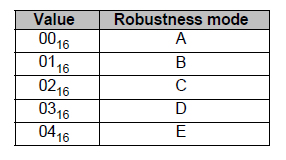


Table 4.1: Robustness mode encoding

NOTE: The value of the "robm" TAG Item may be checked to identify whether RSCI Packets are scheduled to be transmitted every 400 ms (robustness modes A to D) or every 100 ms (robustness mode E).

**4.2 FAST ACCESS CHANNEL (fac\_)**

This TAG item as shown in figure 4.2 holds the complete FAC information as transmitted in the DRM multiplex and shall be included in every RSCI TAG packet. If no FAC information is available an empty TAG item shall be transmitted. This TAG item is mandatory for all RX\_STAT profiles except profile B.

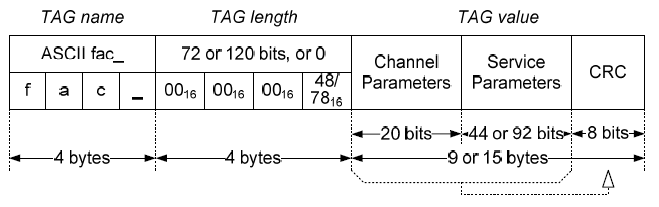
****

Figure 4.2: Fast Access Channel

* Channel parameters:channel parameter section of the FAC as described in chapter 3.2.
* Service parameters:service parameter section of the FAC as described in chapter 3.2 (the data carried in the service parameters section shall be repeated according to the FAC repetition rules described in chapter 3.4). The length of this section depends on the robustness mode used: 44 bits (1 service description) for robustness modes A to D, or 92 bits (2 service descriptions of 44 bits each plus 4 padding bits set to 0) for robustness mode E.
* CRC:checksum over the previous part of the TAG value as described in chapter 3.4.

**4.3 SERVICE DESCRIPTION CHANNEL (sdc\_)**

As shown in figure 4.3 this TAG item holds one complete SDC block as transmitted in the DRM multiplex and shall be included in the TAG packet containing the data for the first logical frame of each super frame. The TAG length of this TAG item in any other TAG packets shall be zero. Because of the data rate this TAG item is mandatory only for RX\_STAT profiles A, C, D and R.

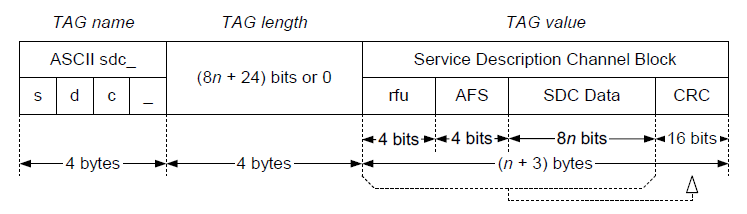
**

Figure 4.3: Service Description Channel

* rfu:these four bits are reserved for future use and shall have the value zero.

NOTE : Bits 0 to 3 (most significant bits of the first byte of the TAG value which are bit positions 4 to 7 of the first byte) are padded with zeros to keep the byte alignment of the rest of the SDC data block.

* Alternative Frequency Switching (AFS):The **AFS index** is an unsigned binary number in the range 0 to 15 that indicates the number of transmission superframes which separate this SDC block from the next with identical content when the identity field in the FAC is set to 00. The AFS index shall be identical for all SDC blocks. The AFS index may be changed at reconfiguration.
* SDC Data:data block of the SDC.
* CRC:field shall contain a 16-bit CRC calculated over the AFS index coded in an 8-bit field (4 msbs are 0) and the data field.

The size of the SDC data block (value of *n*) depends upon the robustness mode, constellation diagram used for SDC cells and spectrum occupancy of the DRM ensemble table 9 which lists values in the range of 13 to 207.

NOTE : If an SDC block is present in the actual transmission frame then the TAG length is calculated by (1 byte + SDC data size (in bytes) + 2 bytes CRC) otherwise the TAG length is zero.

**4.4 MSC STREAM DATA <N> (str0, str1, str2 and str3)**

The TAG items "str0", "str1", "str2" and "str3" shall contain the MSC data for the corresponding DRM stream as shown in figure 4.4. If the TAG length is zero, the TAG item may be omitted from the TAG packet. Because of data rate reasons these TAG items are mandatory only for RX\_STAT profiles A, D and R.

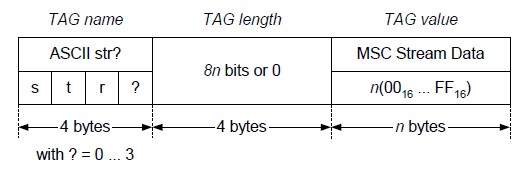
****

Figure 4.4: MSC stream data

The specific MSC stream is referred by use of the TAG name "str0", "str1", "str2" or "str3" appropriate to the stream Id.

* MSC stream data:the content of one specific MSC stream present in the DRM multiplex.

The TAG length is the size of the MSC stream data in bits transported within one multiplex frame.

**CHAPTER 5**

**DIRECT MEMORY ACCESS**

**5.1 INTERNAL DMA (IDMA):**

The internal DMA (IDMA) controller allows rapid data transfers between all local memories. It allows a fast way to page data sections to any memory-mapped RAM local to the CPU. The key advantage of the IDMA is that it allows for transfers between slow (L2) and fast (L1D) data memory, which provides lower latency than the cache controller. In addition, the transfers take place in the background of CPU operation, so that stalls due to cache can be removed from the system.

The IDMA consists of two channels. The two channels are fully orthogonal to each other allowing concurrent operation. The channels are:

* IDMA0 is intended for quick programming of configuration registers located in the external configuration space.
* IDMA1 is intended for transferring data between local memories.

**5.2 ENHANCED DIRECT MEMORY ACCESS (EDMA)**

The enhanced direct memory access (EDMA3) controller’s primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. Typical usage includes, but is not limited to the following:

* Servicing software-driven paging transfers (e.g., transfers from external memory, such as DDR2 to internal device memory, such as DSP L2 SRAM).
* Servicing event-driven peripherals, such as a serial port.
* Performing sorting or sub-frame extraction of various data structures.
* Offloading data transfers from the main device CPU(s) or DSP(s) (see the device-specific data manual for specific peripherals that are accessible via the EDMA3 controller.).

The EDMA3 controller consists of two principal blocks:

* EDMA3 channel controller (EDMA3CC).
* EDMA3 transfer controller(s) (EDMA3TC).

The EDMA3 channel controller serves as the user interface for the EDMA3 controller. The EDMA3CC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA3CC serves to prioritize incoming software requests or events from peripherals and submits transfer requests (TRs) to the transfer controller. The EDMA3 transfer controllers are slaves to the EDMA3 channel controller that is responsible for data movement. The transfer controller issues read/write commands to the source and destination addresses that are programmed for a given transfer. The operation is transparent to user.

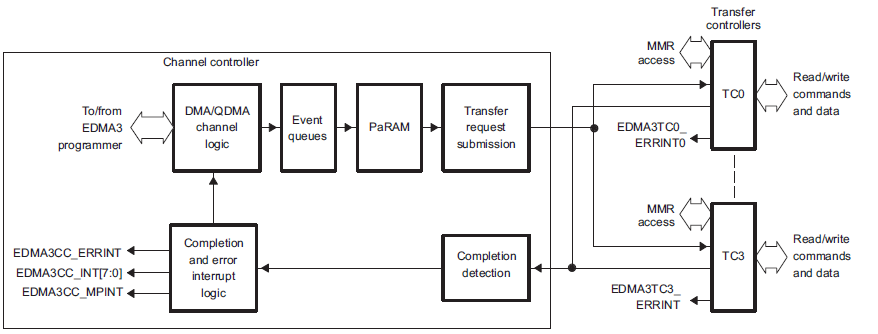
****

Figure 11-1. EDMA3 Controller Block Diagram

**5.3 EDMA3 Channel Controller (EDMA3CC)**

Figure 11-4 shows a functional block diagram of the EDMA3 channel controller (EDMA3CC).

The main blocks of the EDMA3CC are as follows:

* Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. You must write the PaRAM with the transfer context for the desired channels and link parameter sets. EDMA3CC processes sets based on a trigger event and submits a transfer request (TR) to the transfer controller.
* EDMA3 event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
* Completion detection: The completion detect block detects completion of transfers by the EDMA3TC and/or slave peripherals. You can optionally use completion of transfers to chain trigger new transfers or to assert interrupts.
* Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
* Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

Other functions include the following:

* Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA3 programmers own (for example, ARM).
* Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA3CC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels). Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main thing that differentiates a DMA channel from a QDMA channel is the method that the system uses to trigger transfers. See Section 11.3.4.

A trigger event is needed to initiate a transfer. A trigger event may be due to an external event, manual write to the event set register, or chained event for DMA channels. QDMA channels auto-trigger when a write to the trigger word that you program occurs on the associated PaRAM set. All such trigger events are logged into appropriate registers upon recognition.

Once a trigger event is recognized, the appropriate event gets queued in the EDMA3CC event queue. The assignment of each DMA/QDMA channel to an event queue is programmable. Each queue is 16 events deep; therefore, you can queue up to 16 events (on a single queue) in the EDMA3CC at a time. Additional pending events that are mapped to a full queue are queued when the event queue space becomes available. See Section 11.3.11.

If events on different channels are detected simultaneously, the events are queued based on a fixed priority arbitration scheme with the DMA channels being higher priority events than the QDMA channels. Among the two groups of channels, the lowest-numbered channel is the highest priority. PaRAM associated with that channel is read to determine the transfer details. The TR submission logic evaluates the validity of the TR and is responsible for submitting a valid transfer request (TR) to the appropriate EDMA3TC (based on the event queue to the EDMA3TC association, Q0 goes to TC0 , Q1 goes to TC1, Q2 goes to TC2, and Q3 goes to TC3). For more information, refer to Section 11.3.3.

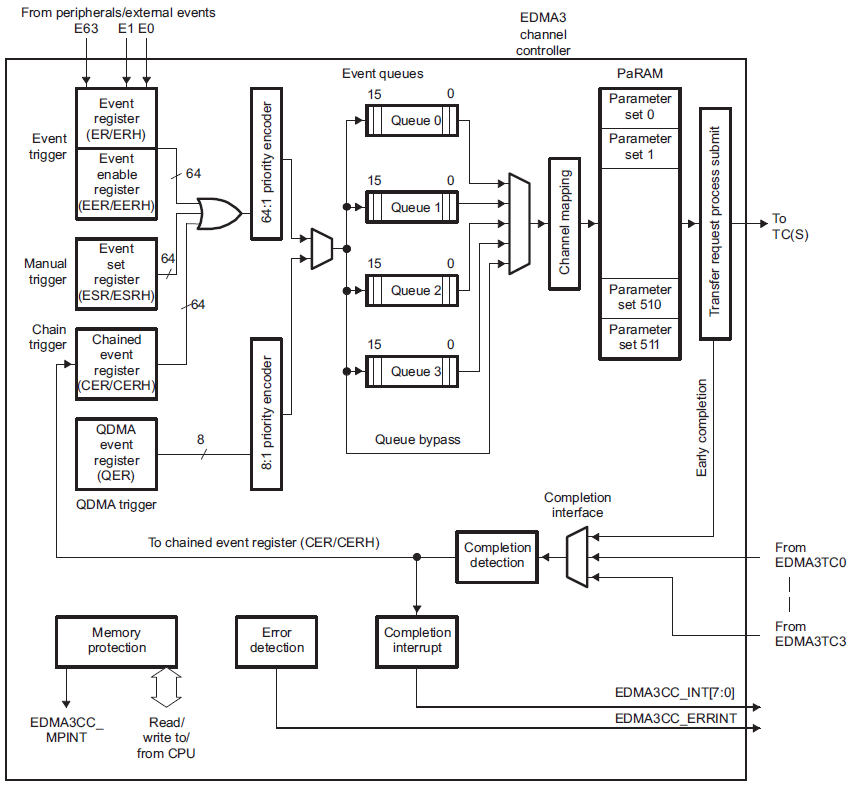


Figure 7-2. EDMA3 Channel Controller (EDMA3CC) Block Diagram

The EDMA3TC receives the request and is responsible for data movement, as specified in the transfer request packet (TRP), other necessary tasks like buffering, and ensuring transfers are carried out in an optimal fashion wherever possible. For more information on EDMA3TC, refer to Section 11.3.1.2.

If you have decided to receive an interrupt or to chain to another channel on completion of the current transfer, the EDMA3TC signals completion to the EDMA3CC completion detection logic when the transfer is complete. You can alternately choose to trigger completion when a TR leaves the EDMA3CC boundary, rather than wait for all of the data transfers to complete. Based on the setting of the EDMA3CC interrupt registers, the completion interrupt generation logic is responsible for generating EDMA3CC completion interrupts to the CPU. For more information, refer to Section 11.3.5.

Additionally, the EDMA3CC also has an error detection logic that causes an error interrupt generation on various error conditions (like missed events, exceeding event queue thresholds, etc.). For more information on error interrupts, refer to Section 11.3.9.4.

**5.4 EDMA3 Transfer Controller (EDMA3TC)**

The main blocks of the EDMA3TC are:

* DMA program register set: The DMA program register set stores the transfer requests received from the EDMA3 channel controller (EDMA3CC).
* DMA source active register set: The DMA source active register set stores the context for the DMA transfer request currently in progress in the read controller.
* Read controller: The read controller issues read commands to the source address.
* Destination FIFO register set: The destination (DST) FIFO register set stores the context for the DMA transfer request(s) currently in progress in the write controller.
* Write controller: The write controller issues write commands/write data to the destination slave.
* Data FIFO: The data FIFO exists for holding temporary in-flight data.
* Completion interface: The completion interface sends completion codes to the EDMA3CC when a transfer completes, and generates interrupts and chained events (also, see Section 11.3.1.1 for more information on transfer completion reporting).

When the EDMA3TC is idle and receives its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA3CC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

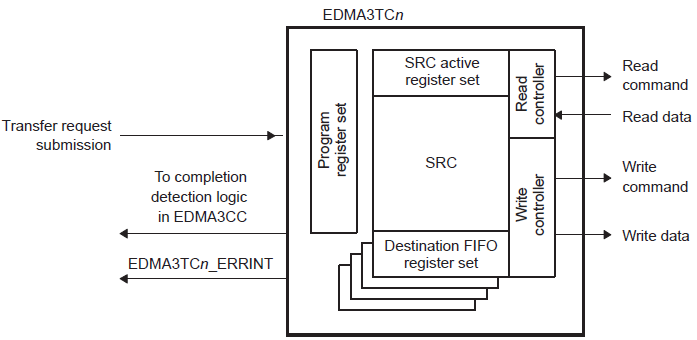


Figure 11-5. EDMA3 Transfer Controller (EDMA3TC) Block Diagram

The read controller issues read commands governed by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization. For more information on command fragmentation and optimization, refer to Section 11.3.12.1.1.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

**5.5 Types of EDMA3 Transfers**

An EDMA3 transfer is always defined in terms of three dimensions. Figure 11-6 shows the three dimensions used by EDMA3 transfers. These three dimensions are defined as:

* 1st Dimension or Array (A): The 1st dimension in a transfer consists of ACNT contiguous bytes.
* 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using SRCBIDX or DSTBIDX.
* 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. Each transfer in the 3rd dimension is separated from the previous by an index programmed using SRCCIDX or DSTCIDX.

Note that the reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (SYNCDIM bit in OPT). Of the three dimensions, only two synchronization types are supported: Asynchronized transfers and AB-synchronized transfers.

**CHAPTER 6**

**MULTICHANNEL AUDIO SERIAL PORT**

**6.1 Introduction**

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT). The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes serializers that can be individually enabled to either transmit or receive.

**6.2 Features**

Features of the McASP include:

* Two independent clock generator modules for transmit and receive.
* Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
* Independent transmit and receive modules, each includes:
* Programmable clock and frame sync generator.
* TDM streams from 2 to 32, and 384 time slots.
* Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
* Data formatter for bit manipulation.
* Individually assignable serial data pins.
* Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
* Wide variety of I2S and similar bit-stream format.
* Integrated digital audio interface transmitter (DIT) supports (up to 10 transmit pins):
  + S/PDIF, IEC60958-1, AES-3 formats.
  + Enhanced channel status/user data RAM.
* 384-slot TDM with external digital audio interface receiver (DIR) device.
* For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
* Extensive error checking and recovery.
  + Transmit underruns and receiver overruns due to the system not meeting real-time requirements.
  + Early or late frame sync in TDM mode.
  + Out-of-range high-frequency master clock for both transmit and receive.
  + External error signal coming into the AMUTEIN input.
  + DMA error due to incorrect programming

**6.3 Protocols Supported**

The McASP supports a wide variety of protocols.

* Transmit section supports:
* Wide variety of I2S and similar bit-stream formats.
* TDM streams from 2 to 32 time slots.
* S/PDIF, IEC60958-1, AES-3 formats.
* Receive section supports:
  + Wide variety of I2S and similar bit-stream formats.
  + TDM streams from 2 to 32 time slots.
  + TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections may each be individually programmed to support the following options on the basic serial protocol:

* Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
* Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
* Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length.
* First-bit data delay: 0, 1, 2 bit clocks.
* Left/right alignment of word inside slot.
* Bit order: MSB first or LSB first.

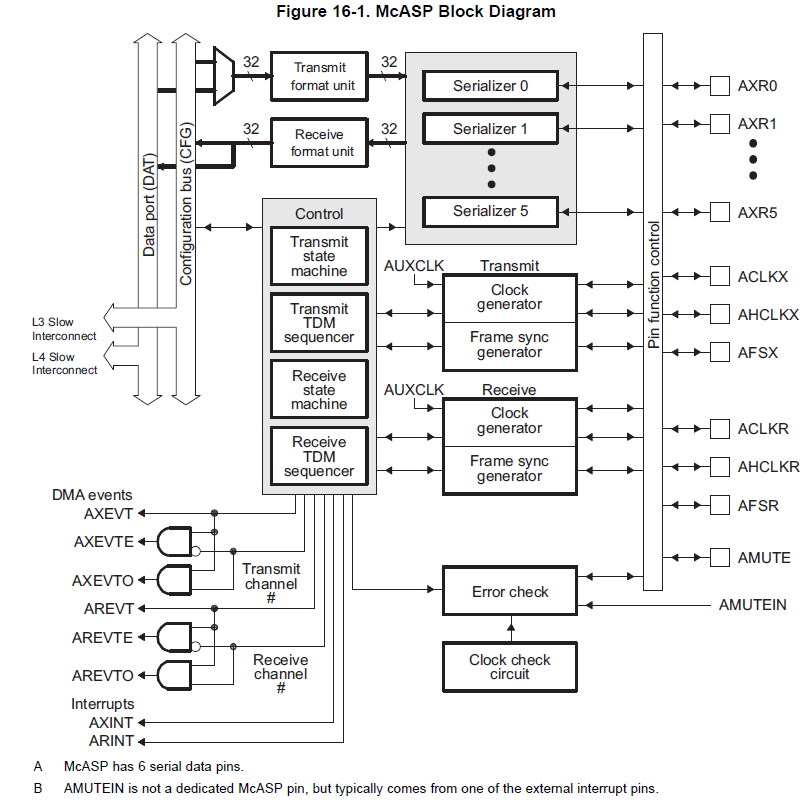
****

Figure 16-1. McASP Block Diagram

* Bit mask/pad/rotate function.
* Automatically aligns data internally in either Q31 or integer formats.
* Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit).
* In DIT mode for McASP, additional features of the transmitter are:
  + Transmit-only mode 384 time slots (subframe) per frame.
  + Bi-phase encoded 3.3 V output.
  + Support for consumer and professional applications.
  + Channel status RAM (384 bits).
  + User data RAM (384 bits).
  + Separate valid bit (V) for subframe A, B.

In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels. In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

**6.4 Format**

The McASP transmitter and receiver support the multichannel, synchronous time-division-multiplexed (TDM) format via the TDM transfer mode. Within this transfer mode, a wide variety of serial data formats are supported, including formats compatible with devices using the Inter-Integrated Sound (I2S) protocol. This section briefly discusses the TDM format and the I2S protocol.

**6.4.1 TDM Format**

The TDM format is typically used when communicating between integrated circuit devices on the same printed circuit board or on another printed circuit board within the same piece of equipment. For example, the TDM format is used to transfer data between the processor and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices. The TDM format consists of three components in a basic synchronous serial transfer: the clock, the data, and the frame sync. In a TDM transfer, all data bits (AXRn) are synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (as defined in Section 16.1.7). The "slots" are also commonly referred to as "time slots" or "channels" in TDM terminology. A frame consists of multiple slots (or channels). Each TDM frame is defined by the frame sync signal (AFSX or AFSR). Data transfer is continuous and periodic, since the TDM format is most commonly used to communicate with data converters that operate at a fixed sample rate. There are no delays between slots.

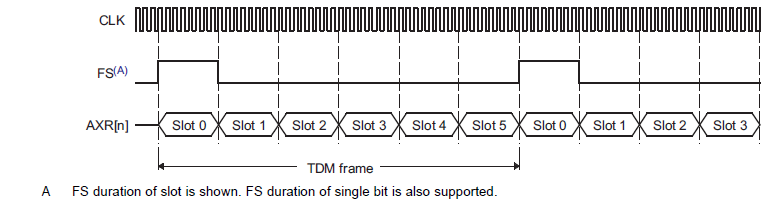
****

Figure 16-7. TDM Format–6 Channel TDM Example

The last bit of slot N is followed immediately on the next serial clock cycle with the first bit of slot N + 1, and the last bit of the last slot is followed immediately on the next serial clock with the first bit of the first slot. However, the frame sync may be offset from the first bit of the first slot with a 0, 1, or 2-cycle delay. It is required that the transmitter and receiver in the system agree on the number of bits per slot, since the determination of a slot boundary is not made by the frame sync signal (although the frame sync marks the beginning of slot 0 and the beginning of a new frame). Figure 16-7 shows the TDM format. Figure 16-8 shows the different bit delays from the frame sync. In a typical audio system, one frame of data is transferred during each data converter sample period fs. To support multiple channels, the choices are to either include more time slots per frame (thus operating with a higher bit clock rate), or to use additional data pins to transfer the same number of channels (thus operating with a slower bit clock rate). For example, a particular six channel DAC may be designed to transfer over a single serial data pin AXRn as shown in Figure 16-7. In this case the serial clock must run fast enough to transfer a total of 6 channels within each frame period.

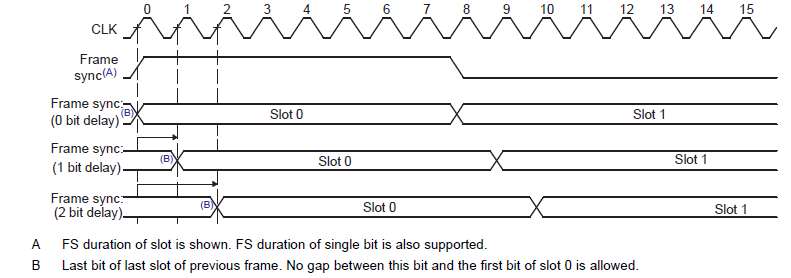
****

Figure 16-8. TDM Format Bit Delays from Frame Sync

Alternatively, a similar six channel DAC may be designed to use three serial data pins AXR[0,1,2], transferring two channels of data on each pin during each sample period. In the latter case, if the sample period remains the same, the serial clock can run three times slower than the former case. The McASP is flexible enough to support either type of DAC.

**6.4.2 Inter-Integrated Sound (I2S) Format**

The inter-integrated sound (I2S) format is used extensively in audio interfaces. The TDM transfer mode of the McASP supports the I2S format when configured to 2 slots per frame. I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXRn. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format is the same as the slot size. The frame signal is also referred to as "word select" in the I2S format. Figure 16-9

shows the I2S protocol. The McASP supports transfer of multiple stereo channels over multiple AXRn pins.

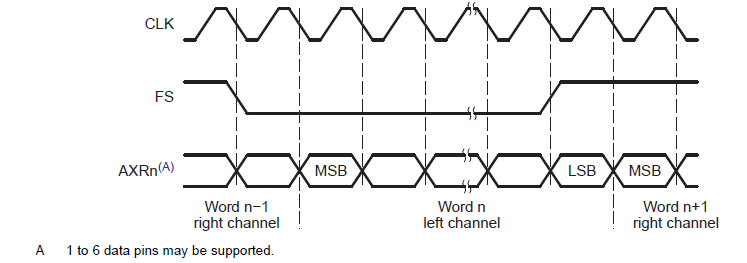
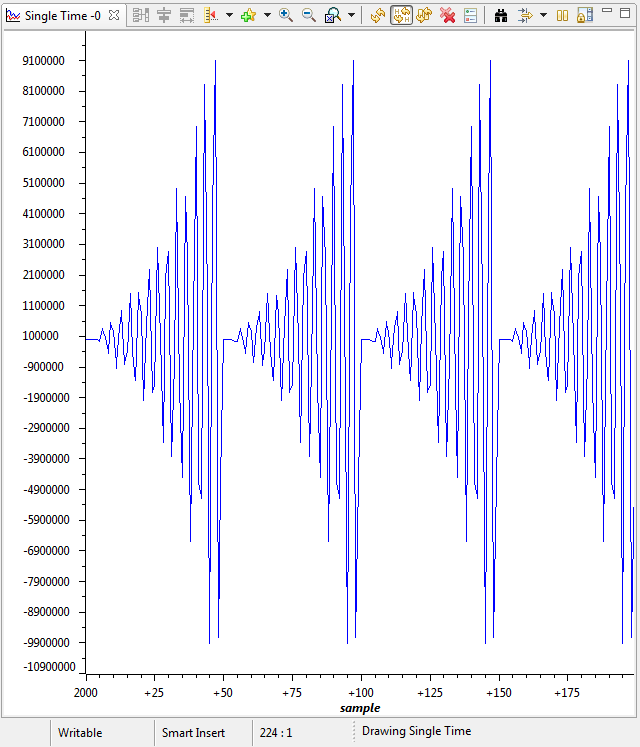


Figure 16-9. Inter-Integrated Sound (I2S) Format

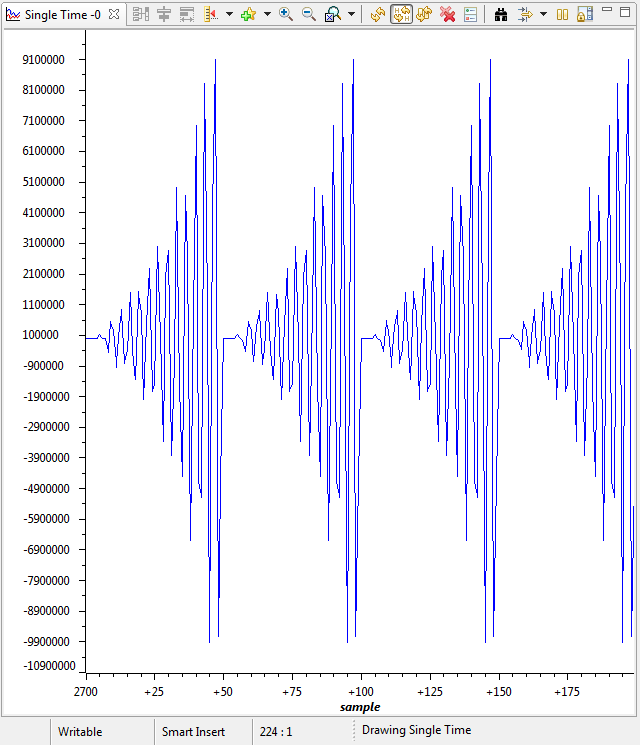
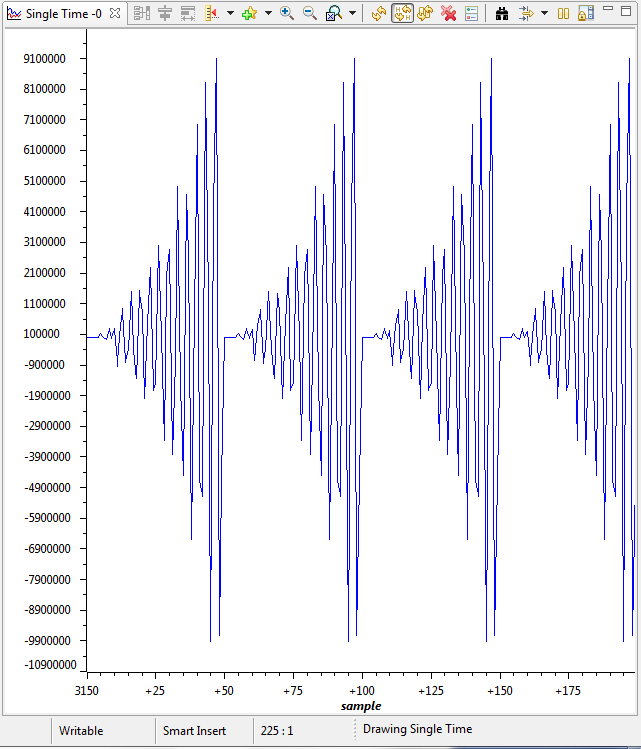
**CHAPTER 7**

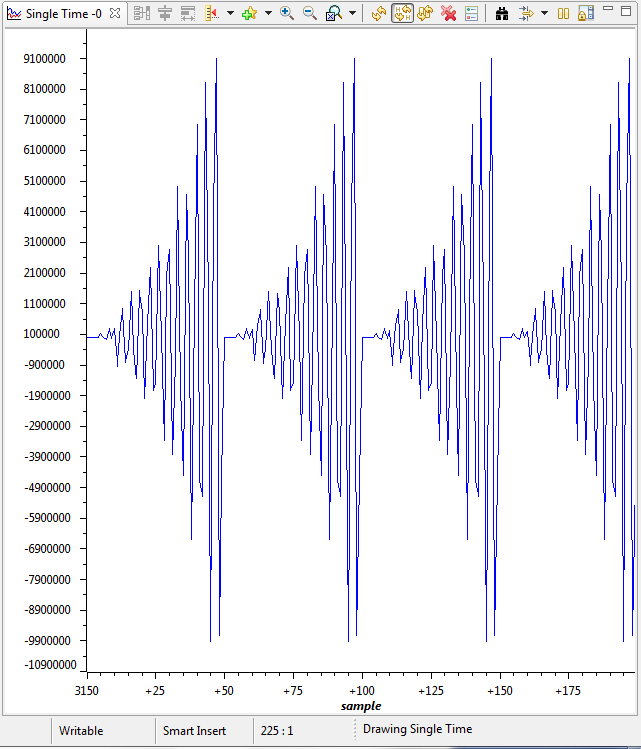
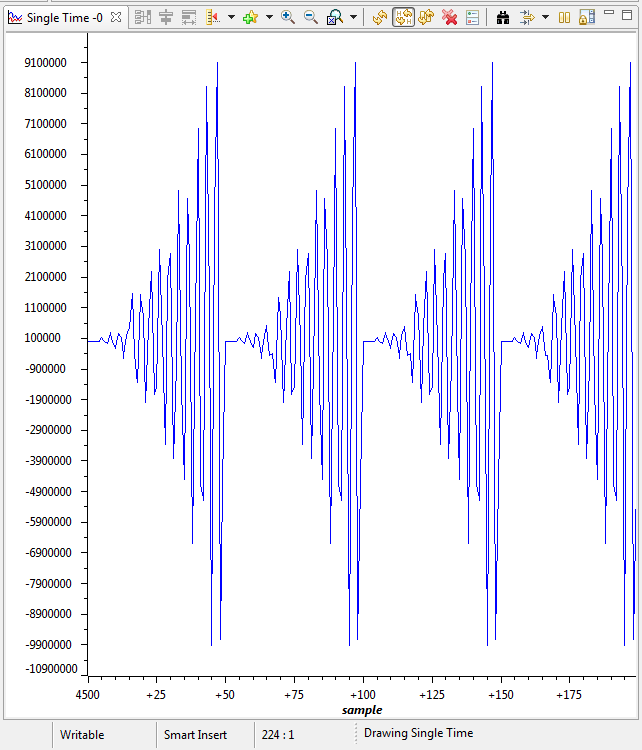
**RESULTS AND DISCUSSION**

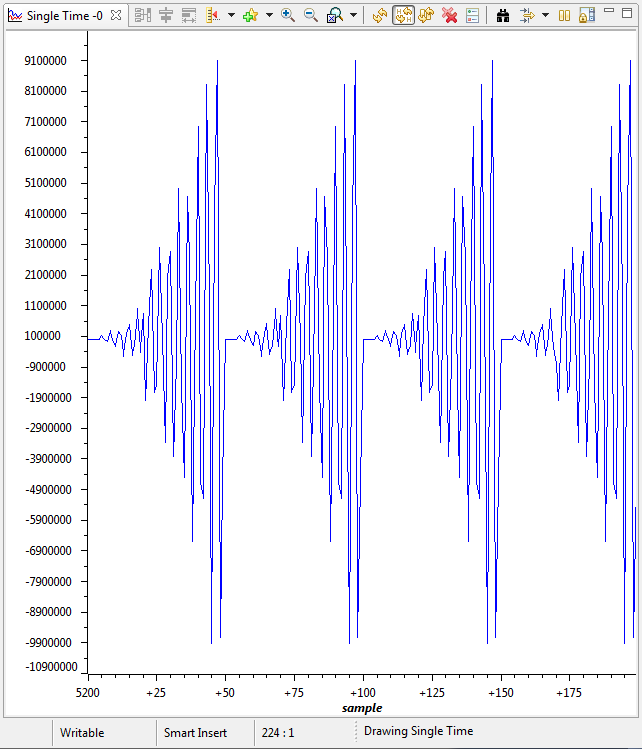
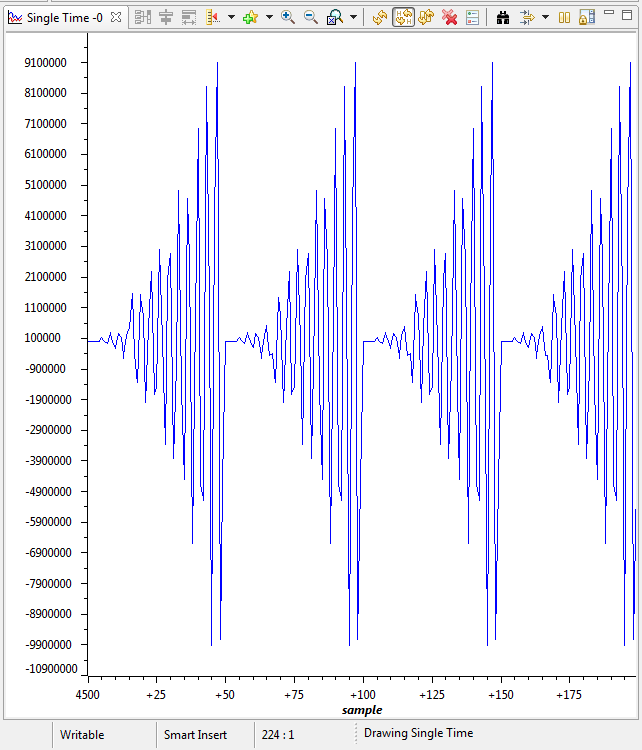
**INPUT WAVEFORM:**

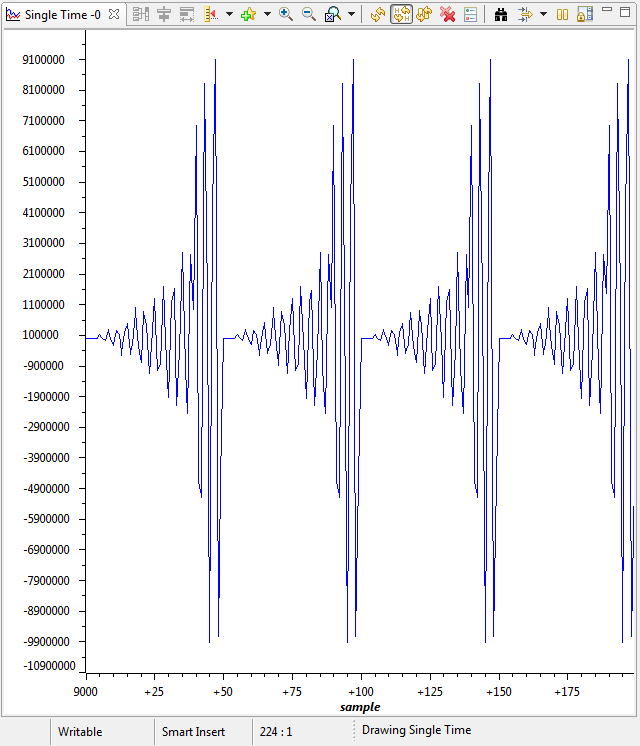
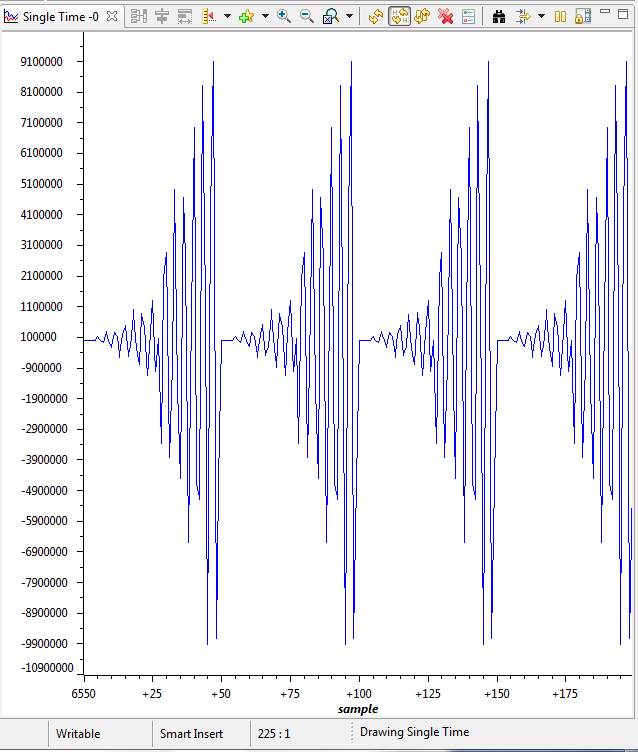
****

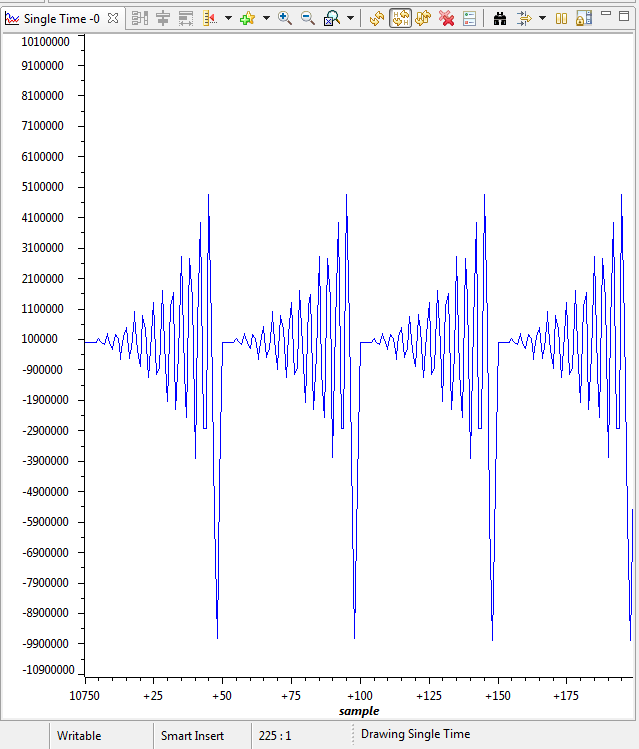
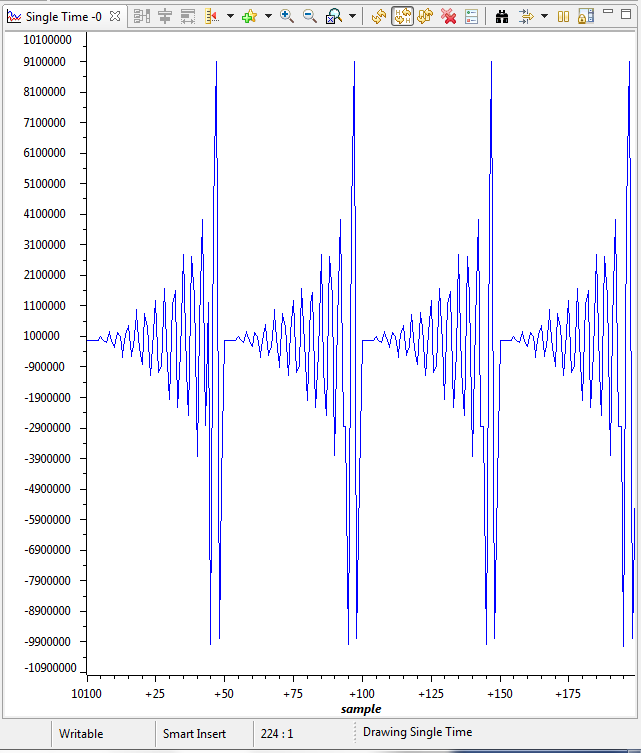
**RESAMPLE:**

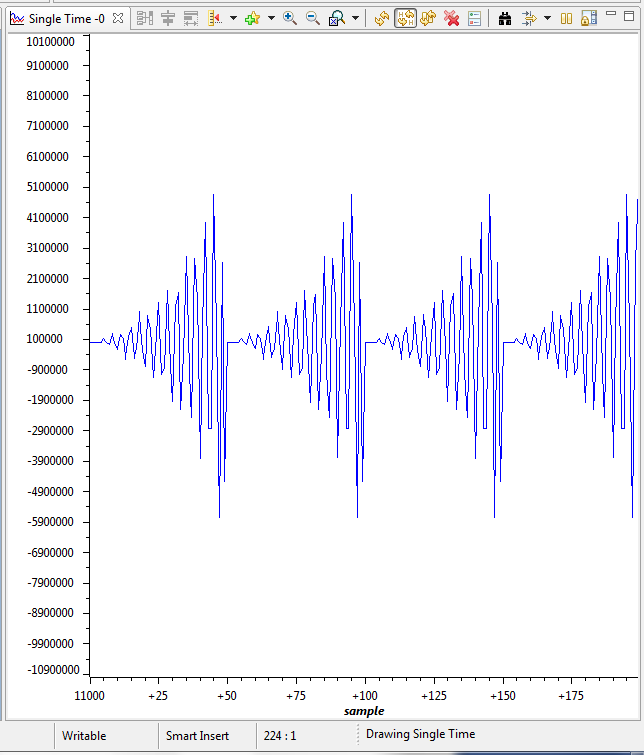
****



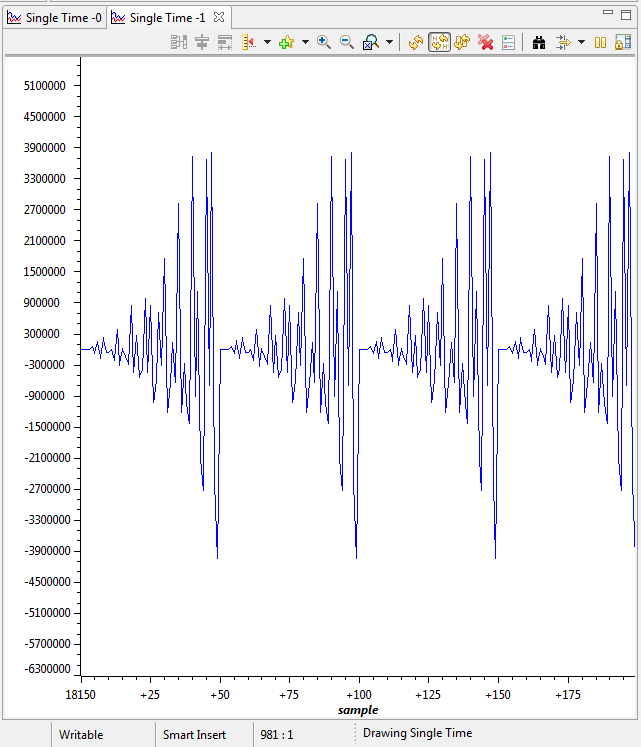




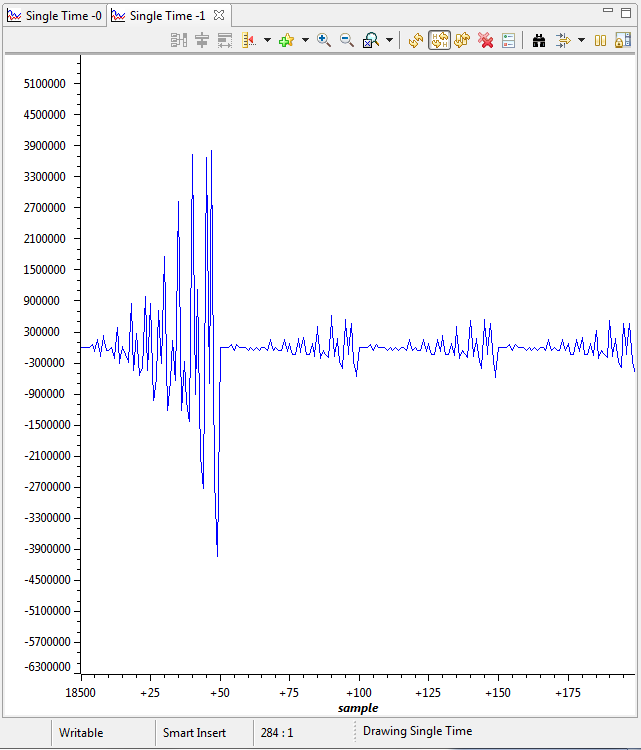
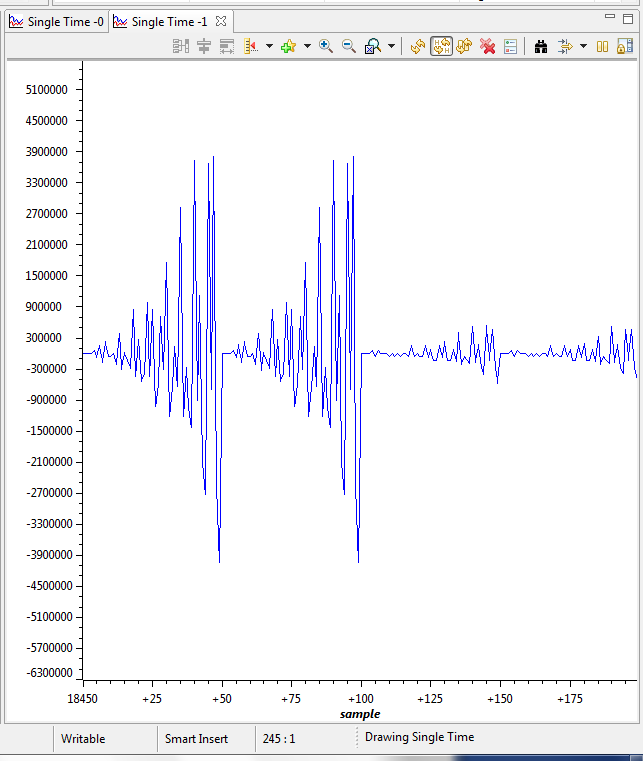




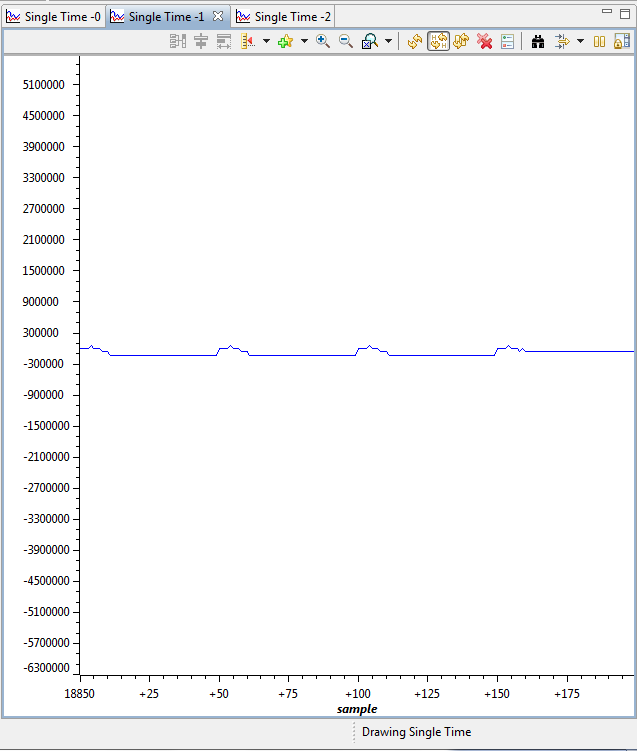
Pre fft process:



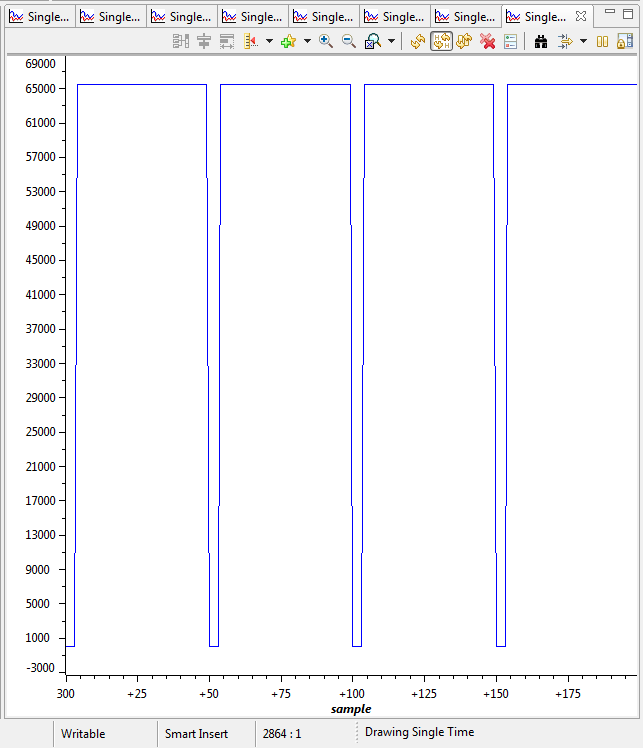
Fft process



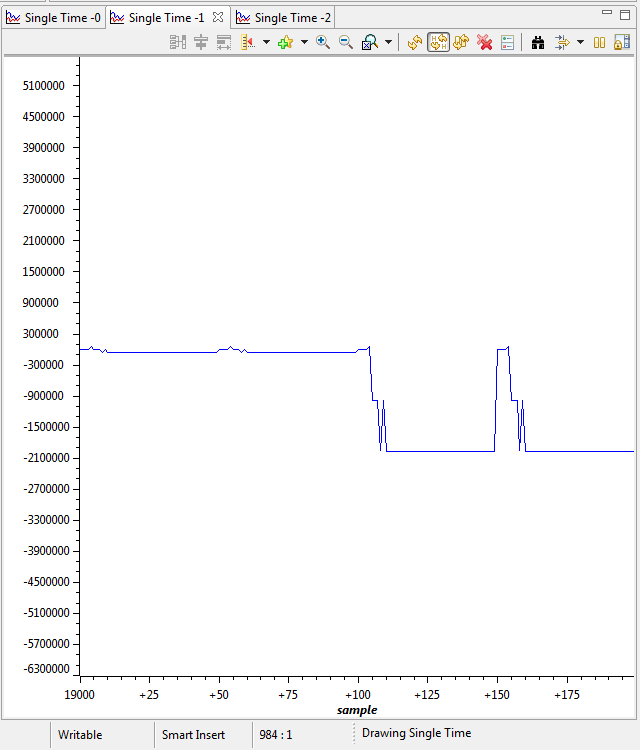
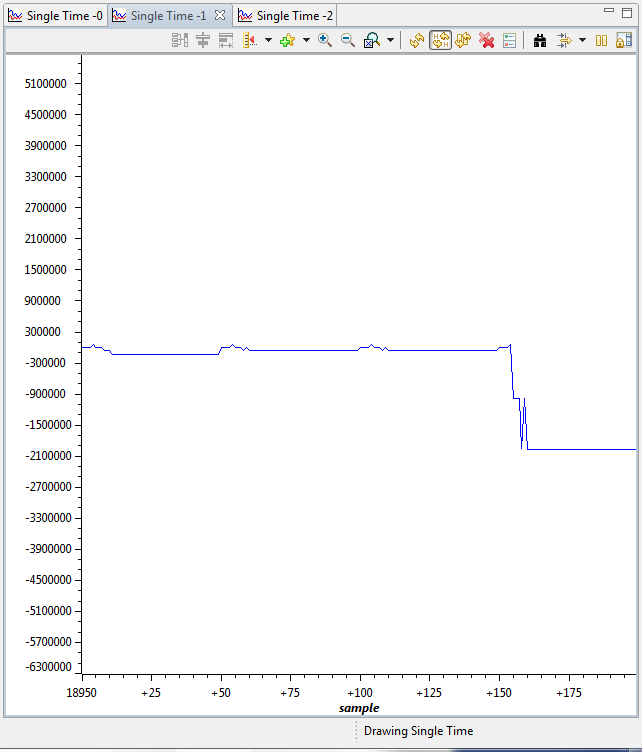
Post fft

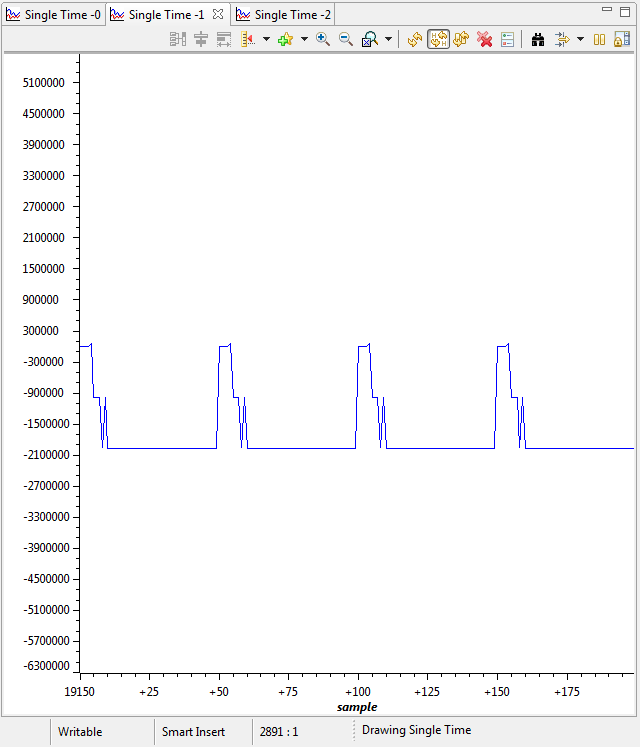
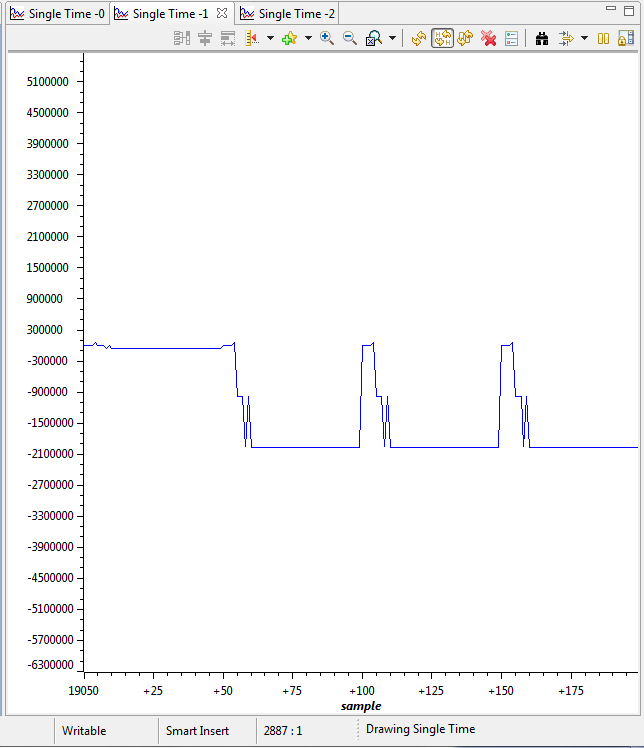


Multiplier value:

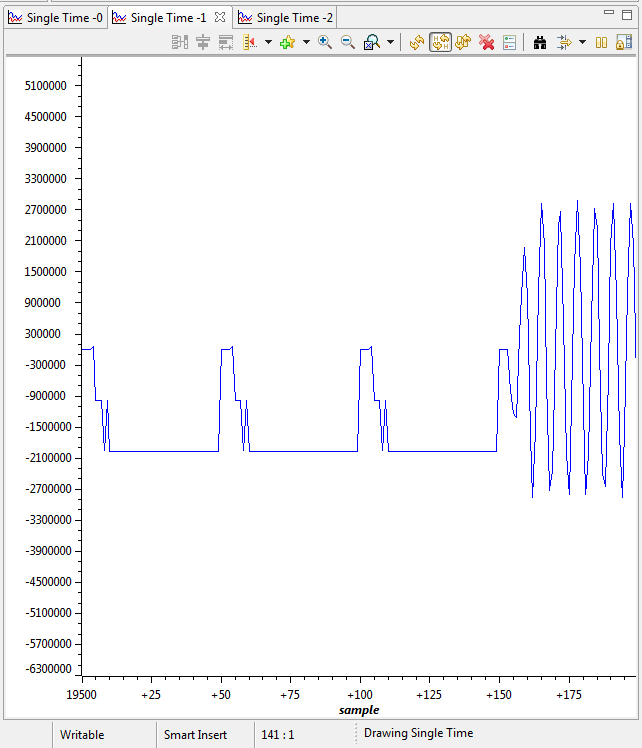


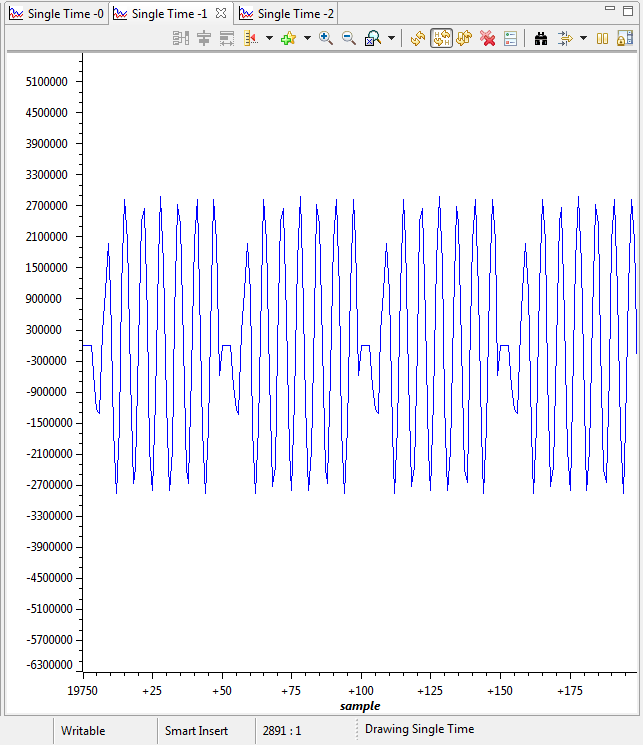
Channel process:

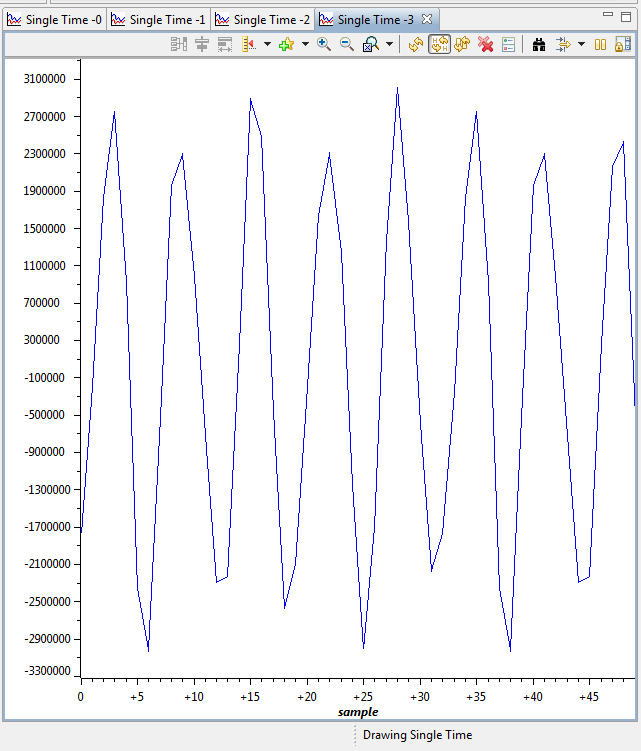




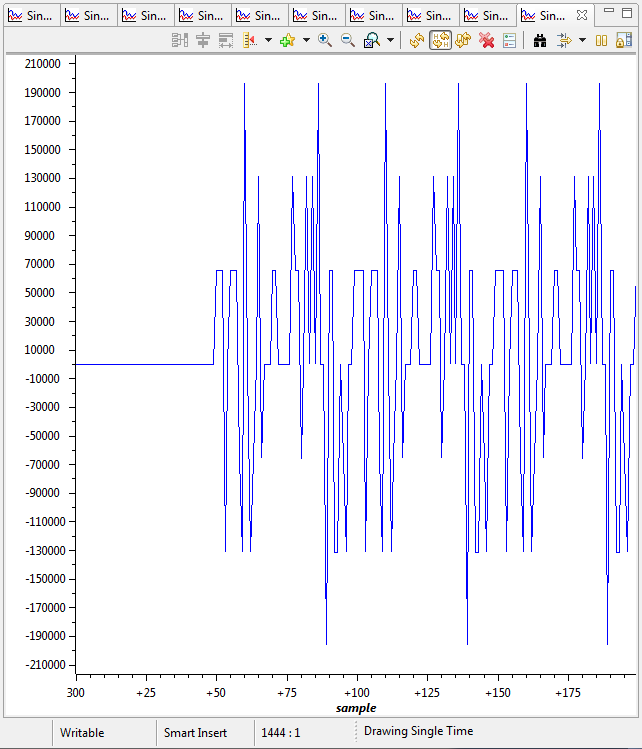
Mlc process:





****

**Decoded output:**

****

**Memory browser:**

**CHAPTER 8**

**CONCLUSION**

This project provides an effective channel decoding technique for the audio files. The channel decoding concept enters a new dimension by using the Jacinto 5 platform as it is a multiprocessor. The core process involved the decoding of the binary bit streams to get the Fast Access Channel (FAC), Service Description Channel (SDC), Main Service Channel (MSC). The process was efficiently handled, by using Viterbi decoder, Biquad filter, Doppler spread, Notch filter.

The working of the Channel decoding was effectively analyzed and tested using Code Composer Studio.