

LA-UR-02-4947

*Approved for public release;
distribution is unlimited.*

Title: DIRECT MEMORY ACCESS FOR THE FIELD- AND
RESONANCE-CONTROL MODULE OF THE SNS
LOW-LEVEL RF CONTROL SYSTEM

Author(s): Sung-il Kown, SNS-02

Submitted to:

Los Alamos

NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

Sung-il Kwon, SNS-2, RF Power Group

1. DIRECT MEMORY ACCESS IN THE FIELD- AND RESONANCE-CONTROL MODULE

In the field-control algorithm, there should be burst data transfer between the external memory interface (EMIF) asynchronous burst SRAM, the digital signal processor (DSP) internal data RAM, the VXI interface dual-port RAM, and the cavity-field dual-port RAM. Because the data stored during the RF pulse ON is massive due to the 20-MHz sampling frequency for each analog signal channel, huge data-memory spaces are necessary. For the 1.1 msec RF ON period in the normal-conducting cavity, we need to sample the cavity field error and to save feedforward-control output for the feedforward-control-output calculation for the next RF pulse. Furthermore, there are eight beam-current profiles and, for each mode, the feedforward-control-output history needs to be stored for the future.

The cavity field error of the current RF pulse period is stored in the cavity dual-port RAM. This data needs to be transferred to the DSP's internal data RAM because the field- and resonance-control module (FRCM) for SNS has two TMS320C6203 DSPs, one for field control and the other for resonance control (frequency control). Both share the same address bus and data bus; therefore, there may be conflicts when both DSPs have access to the data bus because data that is requested by one DSP may be retrieved to the other DSP. Hence, the cavity-field-error data in dual-port RAM must be transferred to the internal RAM via direct memory access (DMA).

For diagnostic purposes, the data stored in the history buffer (FIFO) needs to be transferred to the 16-bit little endian VXI dual-port RAM. Full-size data is not needed for

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

diagnosis (in the case of the 20-MHz and 1.1-msec RF ON period, 22000 samples of each signal will suffice). Instead, decimated data is to be transferred to the VXI dual-port RAM. This decimation and massive data transfer requires DMA.

2. MEMORY MAPS

The Texas Instruments (TI) DSP TMS320C6203 has a 512-kbyte internal data RAM and supports four EMIFs, CE0, CE1, CE2, and CE3. SNS will use the external memory maps shown in the following tables. Table 1 shows the memory map for DSP A (field control) and Table 2 shows the memory map for DSP B (resonance control).

Table 1. DSP A, Field-Control DSP

	Address Range	Size (Bytes)	Descriptions
CE0 32-bit little endian	0x00400000-0x005F FFFF	2M	Asynchronous Burst SRAM
	0x00600000-0x007FFFFFFF	2M	Reserved
	0x00800000-0x013FFFFFFF	12M	Not Accessible
CE1 32-bit little endian	0x01400000-0x015FFFFFFF	2M	Flash Memory
	0x01600000-0x017FFFFFFF	2M	Reserved
CE2 16-bit little endian	0x02000000-0x0200FFFF	64K	VXI Control/Status Registers
	0x02010000-0x0204FFFF	256K	VXI Dual-Port RAM
	0x02050000-0x023FFFFFFF	4M—320K	Reserved
	0x02040000-0x02FFFFFFF	12M	Not Accessible
CE3 32-bit little endian	0x03000000-0x0300FFFF	64K	Control/Status Registers
	0x03010000-0x0304FFFF	256K	DSP B Dual-Port RAM
	0x03050000-0x0308FFFF	256K	NIOS Dual-Port RAM
	0x03090000-0x030AFFFF	128K	Cavity Error PLD Dual-Port RAM
	0x030B0000-0x030CFFFF	128K	Beam PLD Dual-Port RAM
	0x030D0000-0x033FFFFFFF	3M—832K	Reserved
	0x03400000-0x03FFFFFFF	12M	Not Accessible

**Direct Memory Access for the Field- and Resonance-Control Module
of the SNS Low-Level RF Control System**

Table 2. DSP B, Resonance-Control DSP

	Address Range	Size (Bytes)	Descriptions
CE0 32-bit little endian	0x00400000- 0x005F FFFF	2M	Asynchronous Burst SRAM
	0x00600000-0x007FFFFFFF	2M	Reserved
	0x00800000-0x013FFFFFFF	12M	Not Accessible
CE1 32-bit little endian	0x01400000-0x015FFFFFFF	2M	Flash Memory
	0x01600000-0x017FFFFFFF	2M	Reserved
CE2 16-bit little endian	0x02000000-0x0200FFFF	64K	VXI Control/Status Registers
	0x02010000-0x0204FFFF	256K	VXI Dual-Port RAM
	0x02050000-0x023FFFFFFF	4M—320K	Reserved
	0x02040000-0x02FFFFFFF	12M	Not Accessible
CE3 32-bit little endian	0x03000000-0x0300FFFF	64K	Control/Status Registers
	0x03010000-0x0304FFFF	256K	DSP A Dual-Port RAM
	0x03050000-0x0308FFFF	256K	NIOS Dual-Port RAM
	0x03090000-0x030AFFFF	128K	Cavity PLD Dual-Port RAM
	0x030B0000-0x030CFFFF	128K	Forward PLD Dual-Port RAM
	0x030D0000-0x030EFFFF	128K	Reflected PLD Dual-Port RAM
	0x030F0000-0x033FFFFFFF	3M—960K	Reserved
	0x03400000-0x03FFFFFFF	12M	Not Accessible

3. TMS320C6203 DMA STRUCTURE

The TMS320C6203 DSP has four independent DMA channels (shown in Figure 1), and each channel has its own burst FIFO. These are able to co-exist without loss of throughput (1.5 V devices).

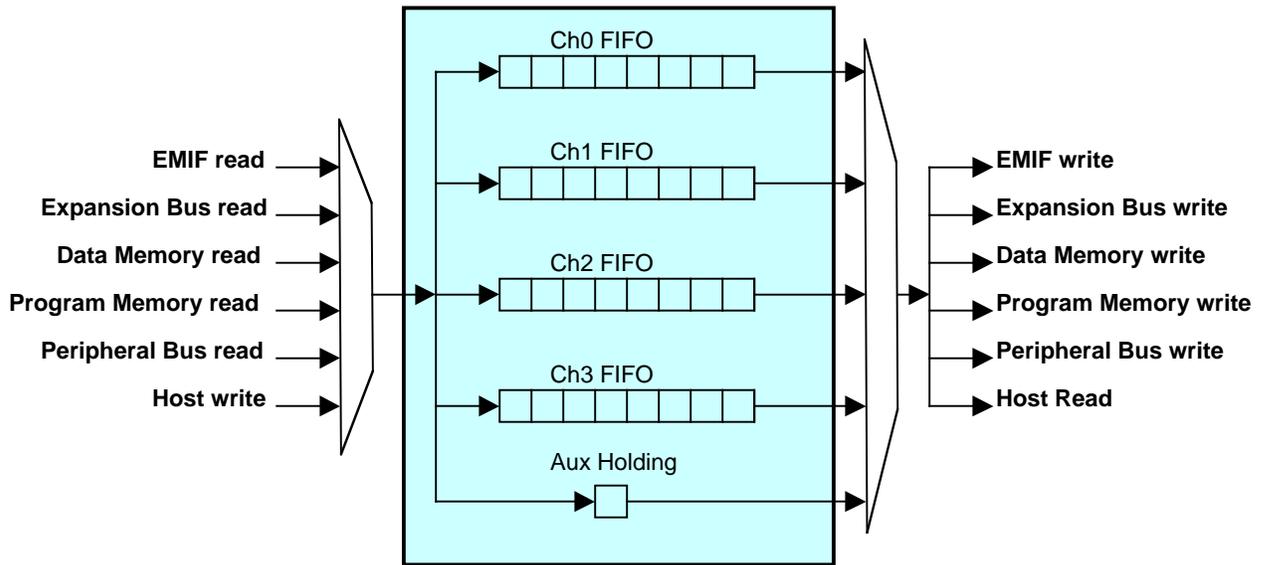


Figure 1. DMA Controller Data Bus Block Diagram for 1.5 V Device

Each channel has a dedicated 9-deep FIFO to facilitate bursting to high-speed memories. Each channel has its own FIFO, which reduces the arbitration required for switching between high-speed bursting channels. The individual operation by any channel is unchanged by any other channel. The benefit of multiple FIFOs comes into play only when switching channels.

4. DMA STRATEGY

4.1 DMA #1, DMA for Cavity Error

The cavity-error data of the RF pulse period, which is stored in the cavity-error dual-port RAM, must be transferred into the Field/Control DSP's internal data memory to update the current feedforward-control output corresponding to the current Pulse ID. The source of the DMA is the dual-port RAM, and the destination is the internal data memory. Figure 2 shows the structure of the I and Q error data stored in the dual-port RAM. The following is the simplest way to achieve DMA:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** increment of address by element size (4 bytes)
- 7) **Source Address:** 0x03090000
- 8) **Destination Address:** DMA_no_1_desBuffer, which is the buffer (array) assigned in the internal data memory

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

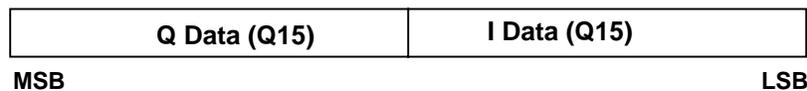


Figure 2. Cavity-Error Dual-Port RAM Data Structure

When the computation of the feedforward-control output has been updated, the transferred data should be decomposed into two Q15-format I error and Q error data.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

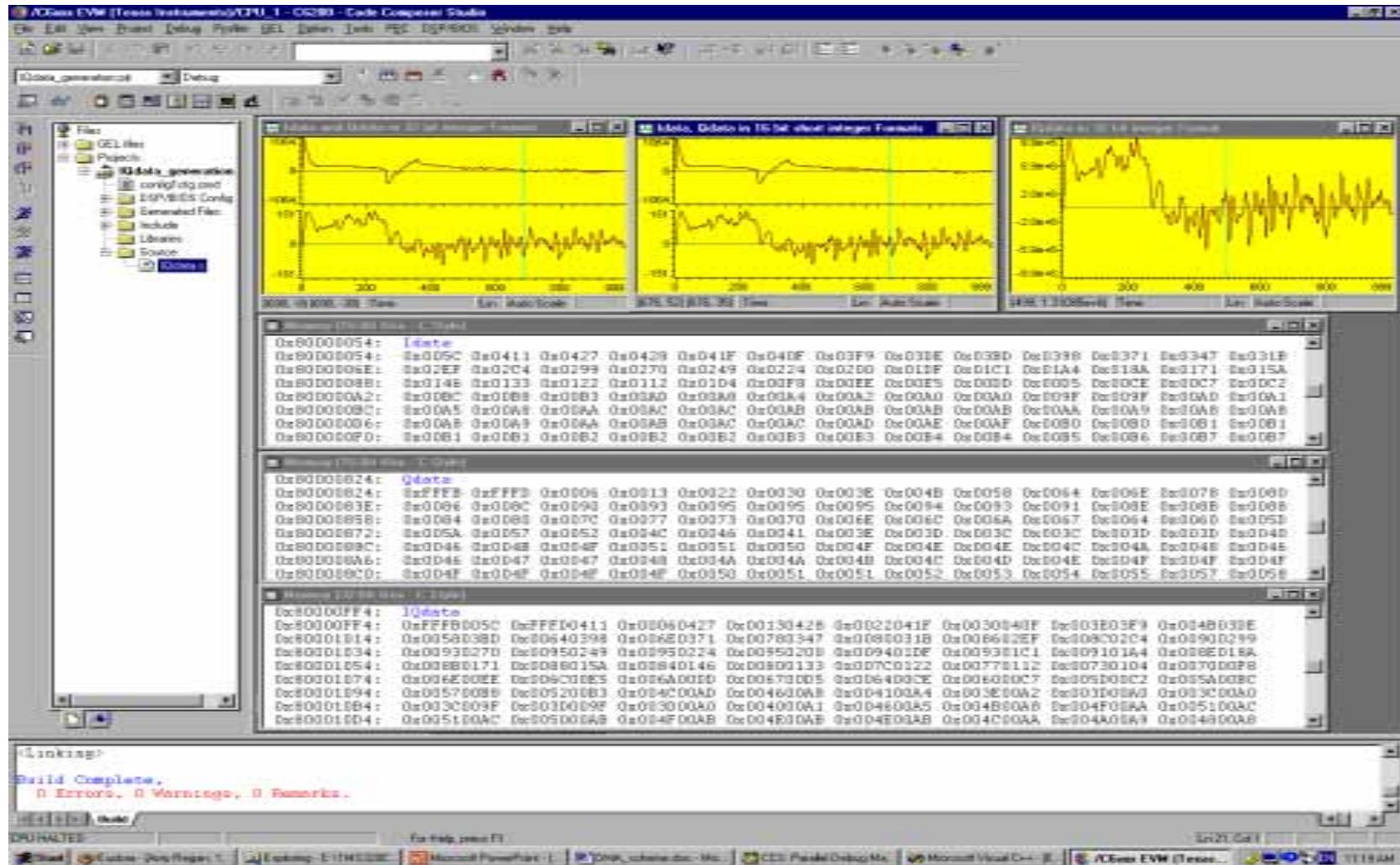


Figure 3. A module, IQCompose(short Idata, short Qdata), is developed to compose one signed integer data from two signed short data. This figure shows the simulation result with the TMS320C6201EVM

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

4.2 DMA #2, DMA for Current Feedforward-Control Output

One beam-current profile, for an RF ON period of 1.1 msec and 20-MHz sampling frequency, requires 22k half-words for I data and 22k half-words for Q data. Thus, the I and Q data together require only 22 words, or 88 kbytes. This is well within the range of the DSP's 512-kbyte internal data memory. However, eight beam-current profiles can require up to 704 kbytes, which exceeds the DSP's memory. As a result, the feedforward-control outputs for eight beam-current profiles, the CE0, are assigned to asynchronous SRAM space (0x00400000-0x005F FFFF, 2M bytes). The memory allocations for eight beam-current profiles are shown in Table 3.

Table 3. Feedforward-Control-Output Table Memory Allocation

Pulse ID	Feedforward Control (32 bits)	Start Address	Size (Bytes)
0	FF0	0x00400000	128 K
1	FF1	0x00420000	128 K
2	FF2	0x00440000	128 K
3	FF3	0x00460000	128 K
4	FF4	0x00480000	128 K
5	FF5	0x004A0000	128 K
6	FF6	0x004C0000	128 K
7	FF7	0x004E0000	128 K

The data structure at each address is the same as that shown in Figure 4, which is compatible with the data structure of the complex programmable logic device (CPLD) registers of the digital low-level RF control system.

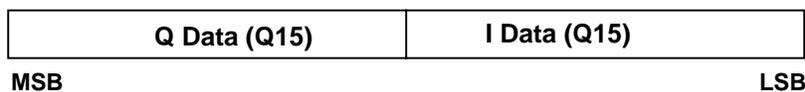


Figure 4. Feedforward-Control-Output Asynchronous SRAM Data Structure

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

To update the feedforward-control output of the current Pulse ID, the data block whose starting address corresponds with the current Pulse ID (as shown in Table 3) is transferred to the DSP's internal data memory. The following are important DMA parameters:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** increment of address by element size (4 bytes)
- 7) **Source Address:** $0x00400000 + (\text{Current Pulse ID}) * 0x00020000$
Current Pulse ID=0,1,2,3,4,5,6,7
- 8) **Destination Address:** DMA_no_2_desBuffer, which is the addresses of a buffer (array) assigned in the internal data memory

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

When the computation of the feedforward-control output is updated, the transferred data should be decomposed into two Q15-format I error and Q error data.

4.3 DMA #3, DMA for Cavity Field 1 History Buffer (FIFO)

The field data for cavity 1 is stored as FIFO 32-K RAM during the RF ON period. For diagnosis, this data is transferred to EPICS through VXI. The data is transferred to the VXI dual-port RAM, and is then retrieved by the host (EPICS). In general, whole data samples are not necessary, and a data decimation is given by the host. The DSP reads the decimation number, which is transferred from the host to a VXI register in the CPLD. This decimation number is used for setting the DMA. The FIFO is a 32-bit little endian and the VXI dual-port RAM is a 16-bit little endian; this determines the element size of the DMA. Because it is impossible to reflect the decimation with FIFO RAM, the data in FIFO is first transferred to the temporary memory space

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

in the DSP's internal data memory and then the DMA is set to reflect the decimation, which is programmed with Global Index Register A or B. Figure 5 illustrates this process.

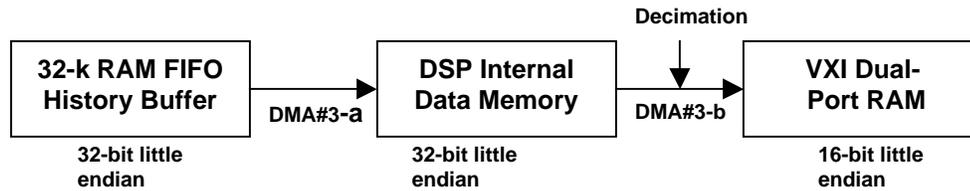


Figure 5. History Buffer Data DMA Scheme (Cavity Field of Cavity 1)

The DMA #3-a has the following parameter setup:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** no modification
- 6) **Destination Direction:** increment of address by element size (4 bytes)
- 7) **Source Address:** 0x0300000C, which is given in CPLD cavity control/status register memory maps
- 8) **Destination Address:** DMA_no_3_desBuffer, which is the buffer (array) assigned in the internal data memory

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

DMA #3-b needs special care. Figure 6 shows an example where 84 32-bit data (I(Q15) and Q(Q15)) are in the DSP's DPS internal data memory and the decimation number is 4 in words.

The following conditions are in effect in this example:

- the element size is to be 16 bits (ESIZE=01b in primary control register),
- the frame count is 21 (=84/4),
- the element count in each frame is 2,
- the frame index, representing the address adjustment in bytes between the start address of the last element of one frame and the start address of the next adjacent frame, is 14(=decimation number*4-2) bytes, and
- the element index representing the address adjustment in bytes between two adjacent elements in a frame (i.e., the address difference in bytes between two start addresses of two adjacent elements in a frame) is 2 bytes.

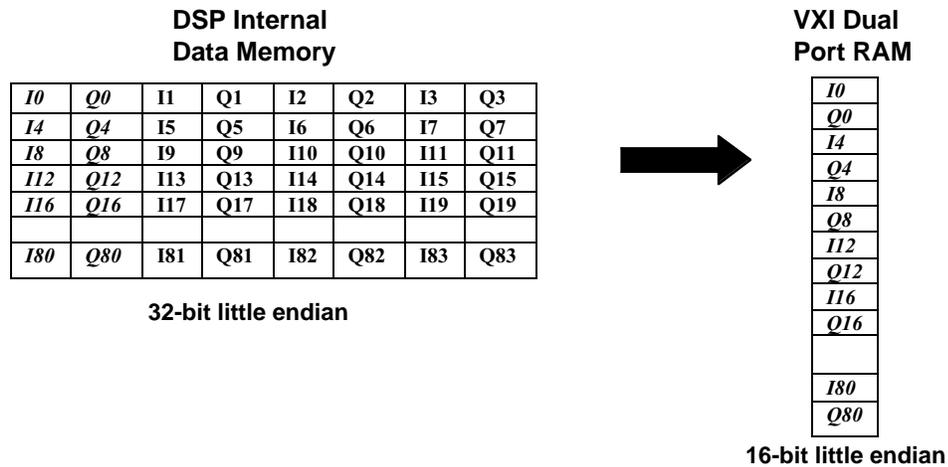


Figure 6. Example of DMA Between the DPS Internal Data Memory and the VXI Dual-Port RAM.

The above example is generalized for the arbitrary number of data in words and the decimation number in words, which results in the following parameter setup (DMA#3-b):

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 01b (16 bit)
- 5) **Source Direction:** 11b, which represents the address modification is programmed by using Global Index Register
- 6) **Destination Direction:** 01b (increment of address by element size [2 bytes])
- 11) **Source Address:** DMA_no_3_desBuffer, which is the buffer (array) assigned in the internal data memory
- 7) **Destination Address:** 0x02010000, which is in CE1 of the DSP B memory maps.
- 8) **Primary Control Register INDEX bit:** 0b, representing Global Index Register A

Transfer Counter Register

- 9) **Frame size:** $(Data\ number\ in\ words)/(Decimation\ number\ in\ words)$
- 10) **Element Count:** 10b (2)

Global Index Register A

- 11) **Frame Index in bytes:** $(Decimation\ number\ in\ words) \times 4 - 2$
- 12) **Element Index in bytes:** 2

4.4 DMA #4, DMA for Cavity Field 2 History Buffer (FIFO)

The same DMA scheme is used, except for the source addresses and the destination addresses (as shown in Figure 7).

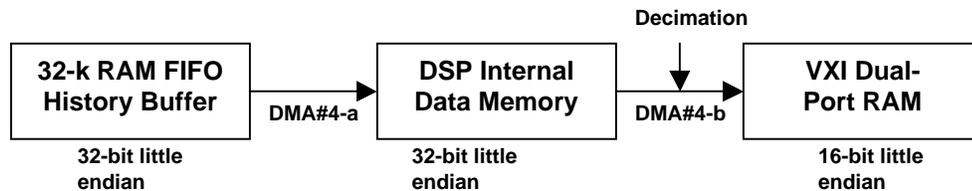


Figure 7. History Buffer Data DMA Scheme (Cavity Field of Cavity 2)

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

DMA #4-a has the following parameter setup:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** increment of address by element size (4 bytes)
- 7) **Source Address:** 0x03000010, which is given in CPLD cavity control/status register memory maps
- 8) **Destination Address:** DMA_no_4_desBuffer, which is the buffer (array) assigned in the internal data memory

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

DMA#4-b has the following parameter setup:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 01b (16 bit)
- 5) **Source Direction:** 11b, which represents the address modification is programmed by using Global Index Register
- 6) **Destination Direction:** 01b (increment of address by element size [2 bytes])
- 7) **Source Address:** DMA_no_4_desBuffer, which is the buffer (array) assigned in the internal data memory
- 8) **Destination Address:** 0x02030000, which is in CE1 of the DSP B memory maps.
- 9) **Primary Control Register INDEX bit :** 0b representing Global Index Register A

Transfer Counter Register

- 11) **Frame size:** (Data number in words)/(Decimation number in word)
- 12) **Element Count:** 10b (2)

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

- 13) *Global Index Register A*
- 14) **Frame Index in bytes:** $(\text{Decimation number in words}) \times 4 - 2$
- 15) **Element Index in bytes:** 2

4.5 DMA #5, DMA for the Update of the Feedforward-Control-Output Table

Figure 8 shows the data structure of the current pulse's feedforward-control module input and output. The inputs to the feedforward-control module are in Q15 format; therefore, the transferred data must first be decomposed into two Q15-format data (I and Q). After computation, they are then composed into 32-bit data, as shown in Figure 8. This decomposition and composition is achieved in the feedforward-control-output computation module.

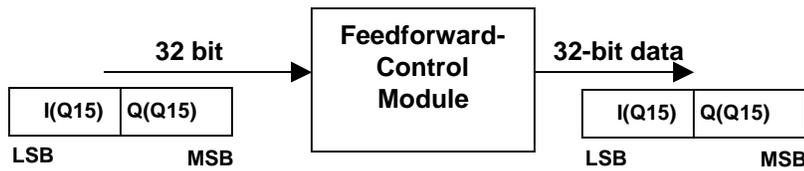


Figure 8. Feedforward-Control Module

After a subroutine computes the feedforward-control output for the future RF pulse, the feedforward-control-output table corresponding to the current Pulse ID is updated depending on whether the current pulse status is a good pulse or a bad pulse. The current pulse status is determined in EPICS, and the determination result is fed back to the DSP (along with the prepulse and the next pulse's Pulse ID) 5 msec (which is programmable) before the next RF pulse is on. (See Figure 9.) When the current pulse status is good, the computed feedforward-control output is transferred from the DSP's internal data memory to the feedforward-control-output table in the asynchronous SRAM, through DMA. When the current pulse status is bad, then the computed data is ignored.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System



Figure 9. Current Pulse Information Fed to the DSP by EPICS

Because the computed result is in the DSP's internal data memory, when the current pulse status is good, the data is transferred from the internal data memory to the asynchronous SRAM, where its start address is determined by the current RF pulse's Pulse ID (see Table 3). The following are important DMA parameters:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** increment of address by element size (4 bytes)
- 1) **Source Address:** The feedforward-control module's output is stored in the buffer (array) assigned in the internal data memory. This buffer is named DMA_no_2_desBuffer when it is an input, and DMA_no_5_srcBuffer when it is an output.
- 7) **Destination Address:** $0x00400000 + (\text{Current Pulse ID}) * 0x00020000$,
Current Pulse ID=0,1,2,3,4,5,6,7

Transfer Counter Register

- 8) **Frame size:** 0
- 9) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

4.6 DMA #6, DMA for Next RF Pulse's Feedforward-Control-Output Load

When the prepulse arrives, along with next RF pulse's Pulse ID and the current RF pulse's status information, the normal operation is to transfer feedforward-control output corresponding to the next RF pulse's Pulse ID to 32-k RAM feedforward FIFO, via DMA.

The following three cases are considered:

- a) **When the current pulse status is good and the next pulse's Pulse ID is the same as the current pulse's Pulse ID.** In this case, the computed feedforward-control output is directly transferred from the DSP's internal data memory to the 32-kbyte feedforward-control-output FIFO. Then, the computed feedforward-control output is transferred, through DMA, from the DSP's internal data memory to the feedforward-control-output table in the asynchronous SRAM to update the feedforward-control-output table.
- b) **When the current pulse status is good and the next pulse's Pulse ID is different from the current pulse's Pulse ID.** In this case, the feedforward-control output corresponding to the next pulse's Pulse ID is transferred from the feedforward-control-output table in asynchronous SRAM to the 32-kbyte feedforward-control-output FIFO. Then, the computed feedforward-control output is transferred, through DMA, from the DSP's internal data memory to the feedforward-control-output table in the asynchronous SRAM to update the feedforward-control-output table.
- c) **When the current pulse status is bad.** In this case, the feedforward-control output corresponding to the next pulse's Pulse ID is transferred from the feedforward-control-output table in asynchronous SRAM to the 32-kbyte feedforward-control-output FIFO. There is no update of the feedforward-control-output table.

The following are important DMA parameters for the data transfer from the DSP internal data memory to the 32-kbyte FIFO (call it DMA#6-a):

Primary Control Register

- 2) CPU interrupt after block move completion
- 3) No read synchronization
- 4) No write synchronization

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

- 5) **Element Size:** 32 bit
- 6) **Source Direction:** increment of address by element size (4 bytes)
- 7) **Destination Direction:** no modification

Source Address: The feedforward-control module's output is stored in the buffer (array) assigned in the internal data memory. This buffer is named DMA_no_2_desBuffer when it is an input, and DMA_no_5_srcBuffer when it is an output.

- 8) **Destination Address:** 0x03000018

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

The following are important DMA parameters for the data transfer from the asynchronous SRAM to the 32-kbyte FIFO (call it DMA#6-b):

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** no modification
- 7) **Source Address:** $0x00400000 + (\text{Next Pulse ID}) * 0x00020000$,
Next Pulse ID=0,1,2,3,4,5,6,7
- 8) **Destination Address:** 0x03000018

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

4.7 DMA #7, DMA for Set-Point-Profile Load

When the single beam-current profile is considered, the set-point profile can be calculated at the initialization module or during the RF pulse OFF period. However, when eight beam-current profiles are considered, memory space availability becomes an issue. The current hardware design shows that the asynchronous SRAM space in CE0 is enough for both the feedforward-control-output table and the set-point-profile table. The same rule is applied to the calculation of the memory assignment for the set-point-profile table as for the feedforward-control-output table. For the 1.1-msec RF pulse ON period and 20-MHz sampling frequency, 88 kbytes are necessary to store I and Q set-point profiles for one beam-current profile. Hence, for eight beam-current profiles, 704 kbytes of memory space is necessary. The second 1 Mbyte of the 2-Mbyte asynchronous SRAM is assigned to store the set-point-profile table, as shown in Table 4.

Table 4. Set-Point-Profile Table Memory Allocation

Pulse ID	Set-Point Profile (32 bit)	Start Address	Size (Bytes)
0	SP0	0x00500000	128 K
1	SP1	0x00520000	128 K
2	SP2	0x00540000	128 K
3	SP3	0x00560000	128 K
4	SP4	0x00580000	128 K
5	SP5	0x005A0000	128 K
6	SP6	0x005C0000	128 K
7	SP7	0x005E0000	128 K

The set-point profiles for eight beam-current profiles are computed at the initialization module and are stored in asynchronous SRAM. The data structure of each address is shown in Figure 10.

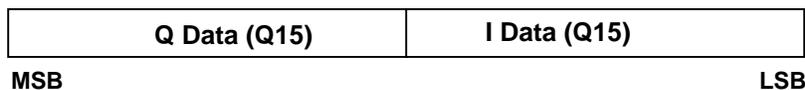


Figure 10. Set Point Profile Asynchronous SRAM Data Structure

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

To compute the set-point profile, we need to know the beam-current amplitude and the beam ON period. This data comes from EPICS. Other information, such as the cavity-filling time and the premargin are fixed. For a normal-conducting cavity, the filling time is 100 μsec and the premargin is 20 μsec . For a superconducting cavity, the filling time is 300 μsec and the premargin is 20 μsec . Figure 11 shows the timing diagram of the set-point profile, the beam current, and the RF gate.

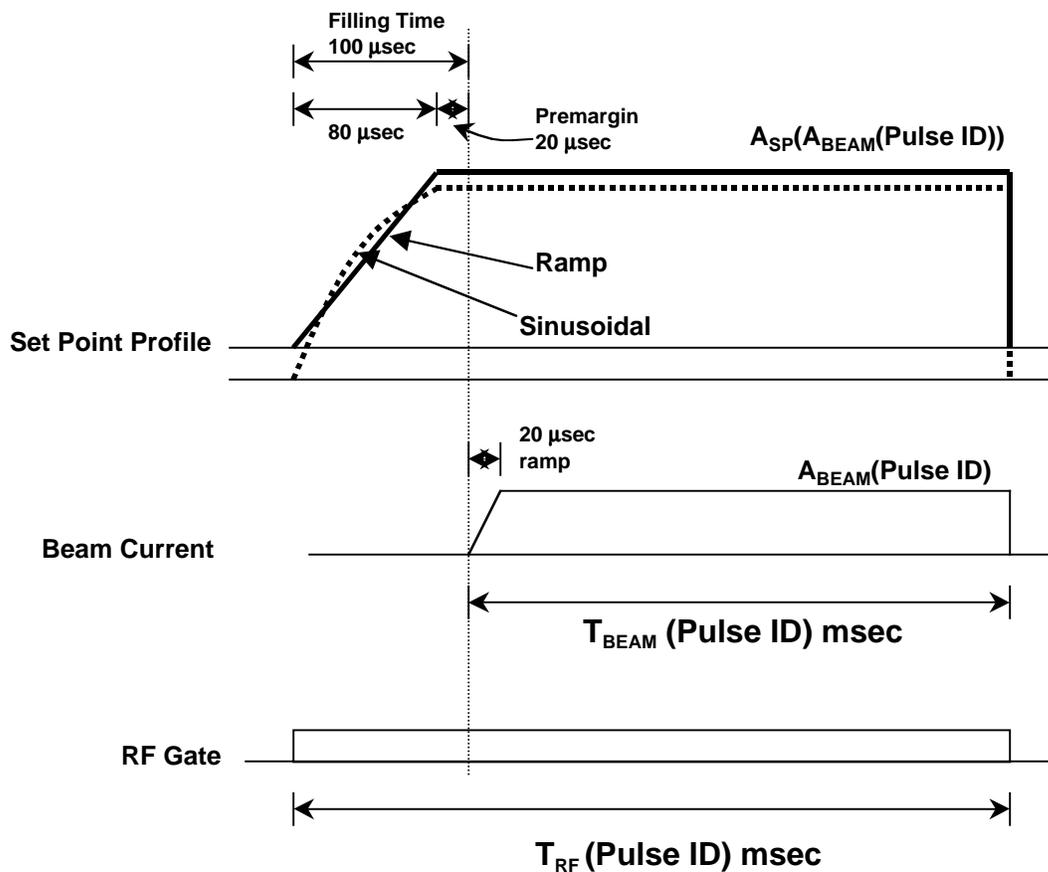


Figure 11. Timing Diagram of the Set-Point Profile, the Beam Current, and the RF Gate. One candidate for the set-point turn-on is ramp, and the other is sinusoidal during the filling time. The beam- pulse period and the RF gate period are functions of Pulse ID. Also, set-point amplitude, A_{SP} , beam-current amplitude, and A_{BEAM} , are functions of Pulse ID.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

To compute the set-point profile with the DSP, the timing diagram of the set-point profile, shown in Figure 11, should be converted to the discrete-time domain. Note that the sampling frequency is 20 MHz. Figure 12 shows the result for the set-point profile with ramp. The magnitude $A_{SP}(A_{BEAM}(Pulse\ ID))$ is represented in Q15 format. The sample numbers N_F , N_{PM} , and $N_{BEAM}(Pulse\ ID)$ correspond to $80\ \mu\text{sec}$, $20\ \mu\text{sec}$, and $T_{BEAM}(Pulse\ ID)$, respectively.

$$N_F = 80e^{-6} \times 20e^6 = 1600 \text{ (samples)}$$

$$N_{PM} = 20e^{-6} \times 20e^6 = 400 \text{ (samples)}$$

$$N_{BEAM}(PulseID) = T_{BEAM}(PulseID) \times 20e^6 \text{ (samples).}$$

Also, the slope of the ramp is

$$slope = \frac{A_{SP}(A_{BEAM}(PulseID))}{N_F}.$$

Based on the above equations, the set-point profile corresponding to Pulse ID is calculated and stored in asynchronous SRAM.

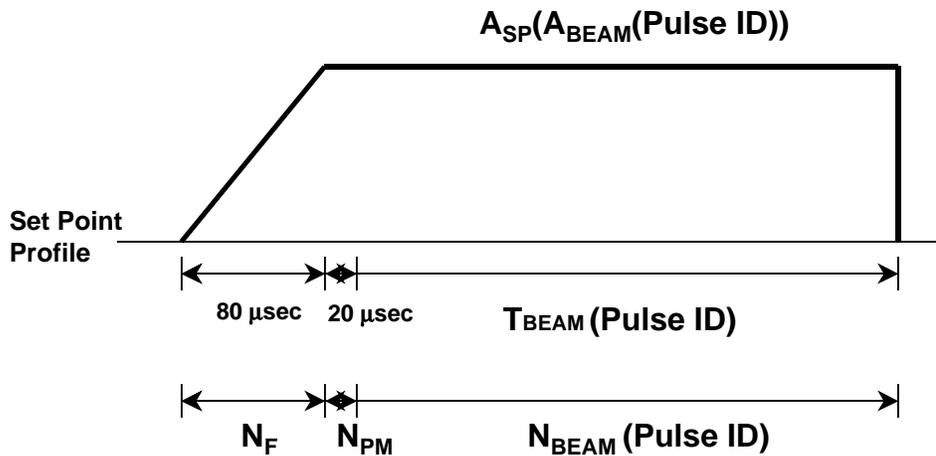


Figure 12. Timing Diagram of the Set-Point Profile in Discrete-Time Domain

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

When the prepulse arrives, along with the Pulse ID, the corresponding set-point profile is transferred, via DMA, to a 32-k RAM set-point FIFO. The following are the important DMA parameters:

Primary Control Register

- 1) CPU interrupt after block move completion
- 2) No read synchronization
- 3) No write synchronization
- 4) **Element Size:** 32 bit
- 5) **Source Direction:** increment of address by element size (4 bytes)
- 6) **Destination Direction:** no modification
- 7) **Source Address:** $0x00500000 + (\text{Next Pulse ID}) * 0x00020000$,
Next Pulse ID=0,1,2,3,4,5,6,7
- 8) **Destination Address:** 0x03000014

Transfer Counter Register

- 9) **Frame size:** 0
- 10) **Element Count:** RF ON period (sec)*Sampling Frequency (Hz). Default value: 22000

5. GROUPING of DMA's to IMPROVE THROUGHPUT

Thus far, several DMA schemes have been addressed. To improve the throughput, the DMA's are grouped appropriately. Note that when the previous pulse status is bad, there is no need to update the feedforward-control-output table, which means that DMA #5 is not necessary. Figure 13 shows the grouping when the previous pulse status is good and the next pulse is also good, Figure 14 shows the grouping when the previous pulse status is good but the next pulse is different, and Figure 15 shows the grouping when the previous pulse status is bad. Table 5 summarizes the various DMA schemes.

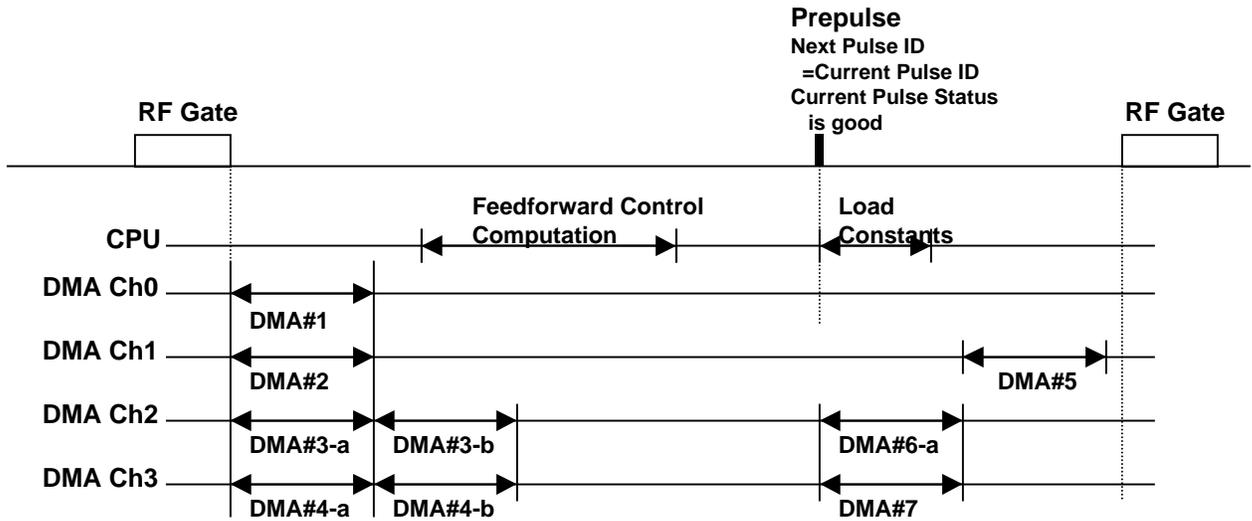


Figure 13. DMA Grouping to Improve Performance When the Previous Pulse is Good and the NextPulse ID is the Same as the Current Pulse ID

**Direct Memory Access for the Field- and Resonance-Control Module
of the SNS Low-Level RF Control System**

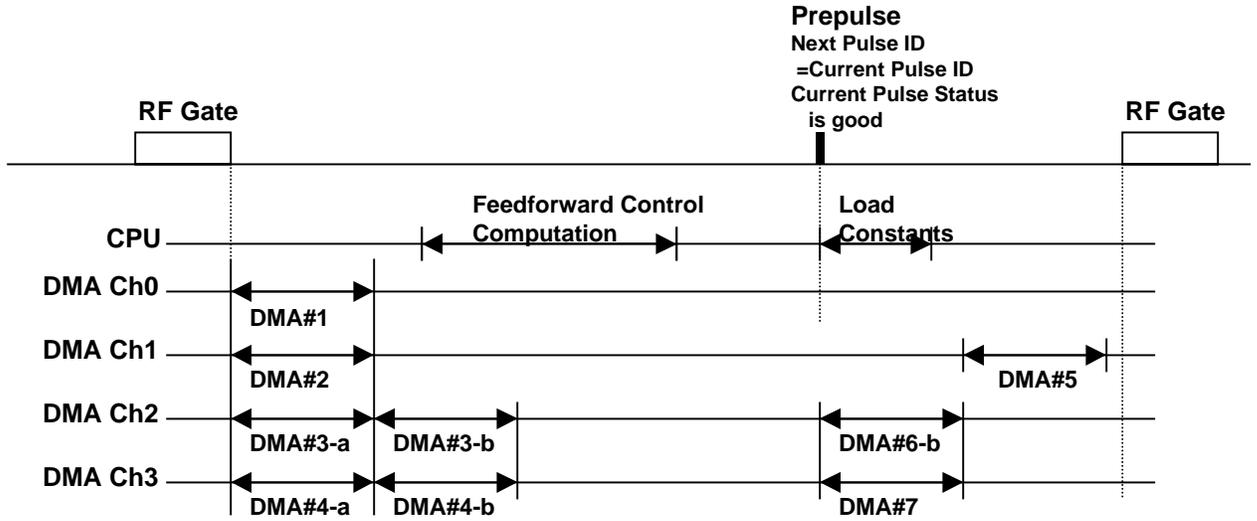


Figure 14. DMA Grouping to Improve Performance when the Previous Pulse is Good and the Next Pulse ID Is Different From the Current Pulse ID

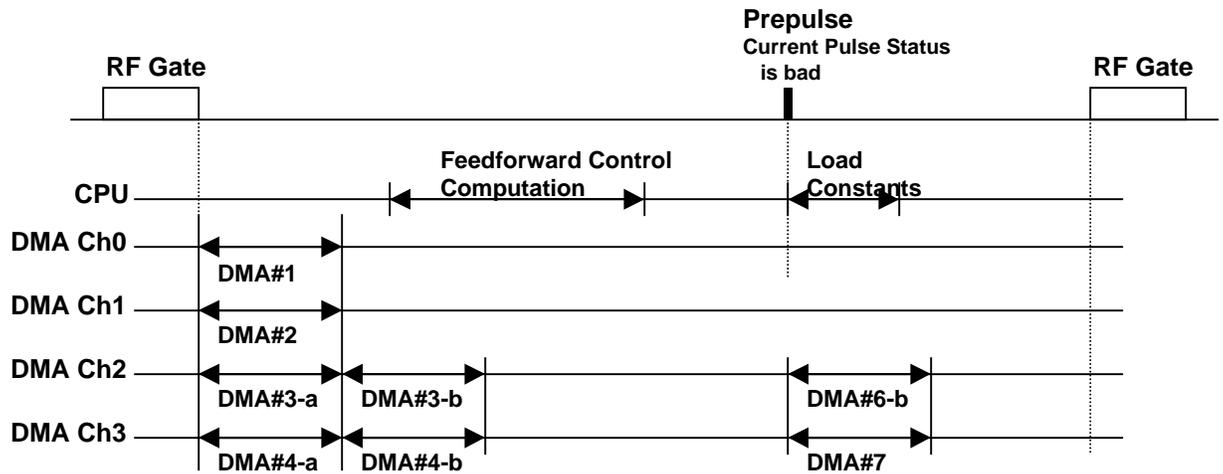


Figure 15. DMA Grouping to Improve Performance when the Previous Pulse is Bad

**Direct Memory Access for the Field- and Resonance-Control Module
of the SNS Low-Level RF Control System**

Table 5. DMA Summary

DMA No.	DMA Channel	Primary Control Register	Secondary Control Register	Transfer Counter Register	Source Address	Destination Address	Global Index Register
1	0	0x03000050	0x00000080	0x0000 ----	0x03090000 (Dual-port RAM)	DMA_no_1_desBuffer (Internal Data Memory)	0x00000000
2	1	0x03000050	0x00000080	0x0000 ----	0x00400000+(Pulse ID)×offset (Async. SRAM)	DMA_no_2_desBuffer (Internal Data Memory)	0x00000000
3-a	2	0x03000040	0x00000080	0x0000 ----	0x0300000C (32 k Cav 1 HB FIFO)	DMA_no_3a_desBuffer (Internal Data Memory)	0x00000000
4-a	3	0x03000040	0x00000080	0x0000 ----	0x03000010 (32k Cav 2 HB FIFO)	DMA_no_4a_desBuffer (Internal Data Memory)	0x00000000
3-b	2	0x03000170	0x00000080	0x00C8 0002	DMA_no_3a_desBuffer (Internal Data Memory)	0x02010000 (VXI Dual-port RAM)	0xYYYY 0002
4-b	3	0x03000170	0x00000080	0x00C8 0002	DMA_no_4a_desBuffer (Internal Data Memory)	0x02030000 (VXI Dual-port RAM)	0xZZZZ 0002
5	1	0x03000050	0x00000080	0x0000 ----	DMA_no_2_desBuffer (Internal Data Memory)	0x00400000+(Pulse ID)×offset(Async. SRAM)	0x00000000
6-a	2	0x03000010	0x00000080	0x0000 ----	DMA_no_2_desBuffer (Internal Data Memory)	0x03000018 (32k FF FIFO)	
6-b	2	0x03000010	0x00000080	0x0000 ----	0x00400000+(Pulse ID)×offset (Async. SRAM)	0x03000018 (32k FF FIFO)	0x00000000
7	3	0x03000010	0x00000080	0x0000 ----	0x00500000+(Pulse ID)×offset (Async. SRAM)	0x03000014 (32k Set Point FIFO)	0x00000000

Pulse ID=0,1,2,3,4,5,6,7

offset=0x00020000

6. PROGRAM STRUCTURE

The c source codes of the DMA data transfers and feedforward-control-output update are shown in Figure 16.

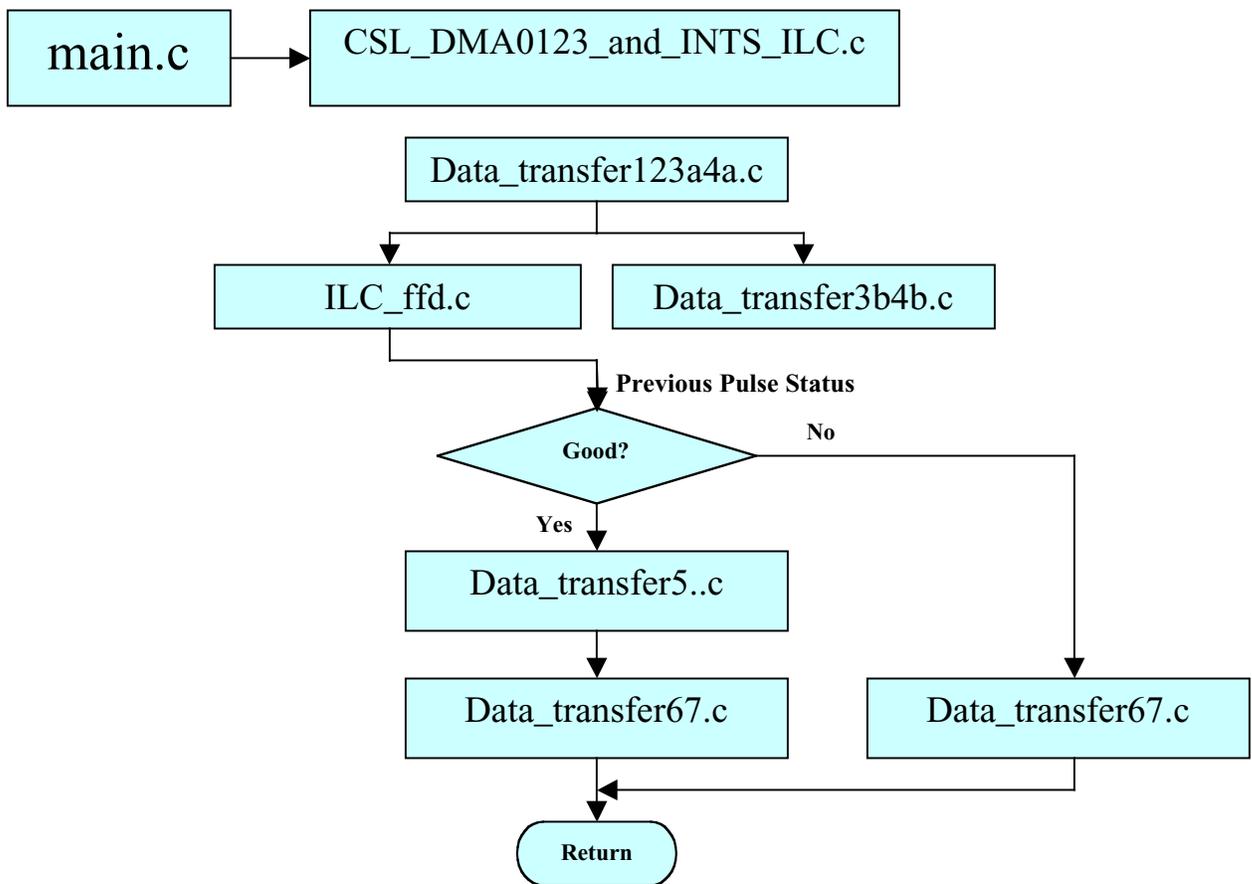


Figure 16. Flowchart of Feedforward-Control-Output Computation

**Direct Memory Access for the Field- and Resonance-Control Module
of the SNS Low-Level RF Control System**

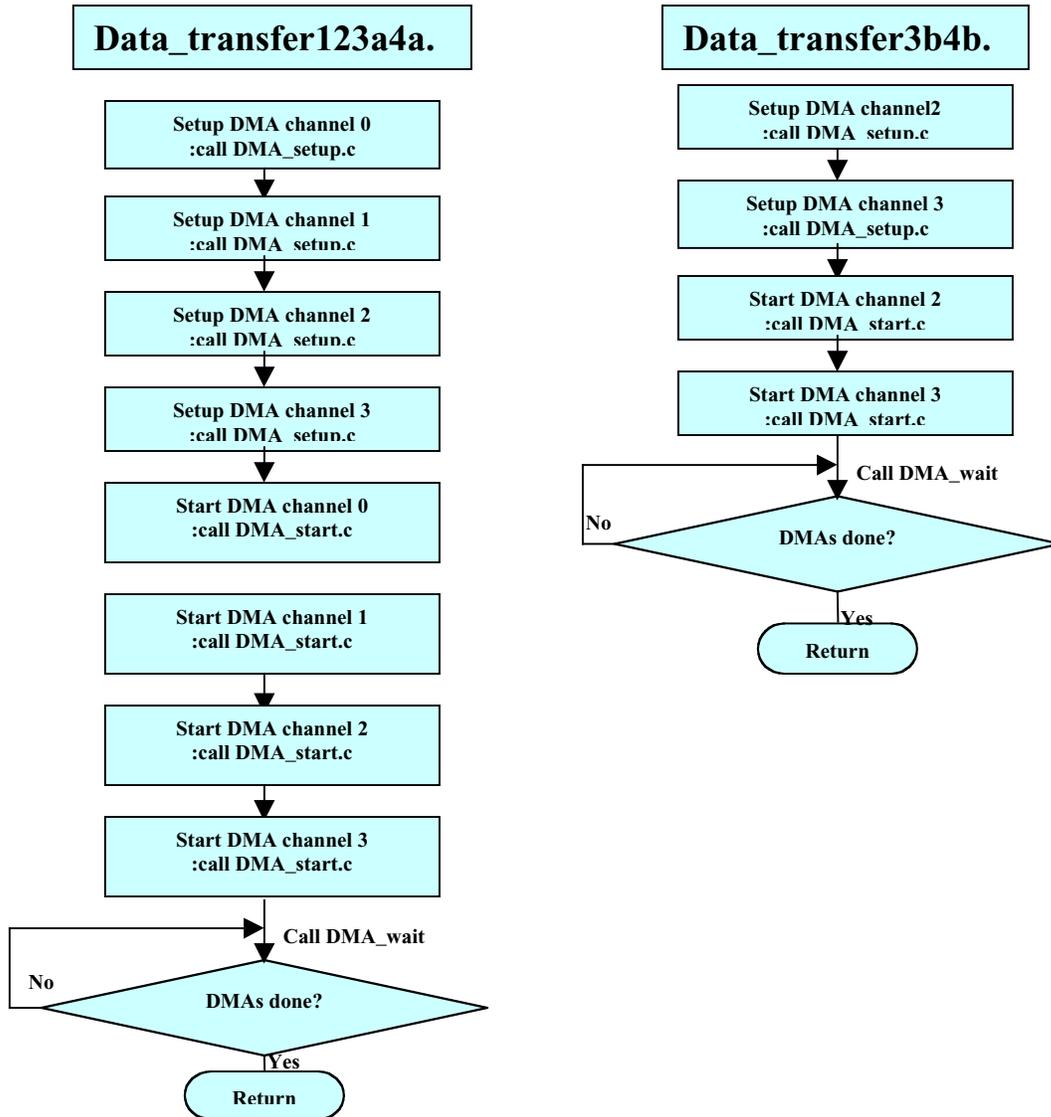


Figure 16. Flowchart of Feedforward-Control-Output Computation (continued)

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

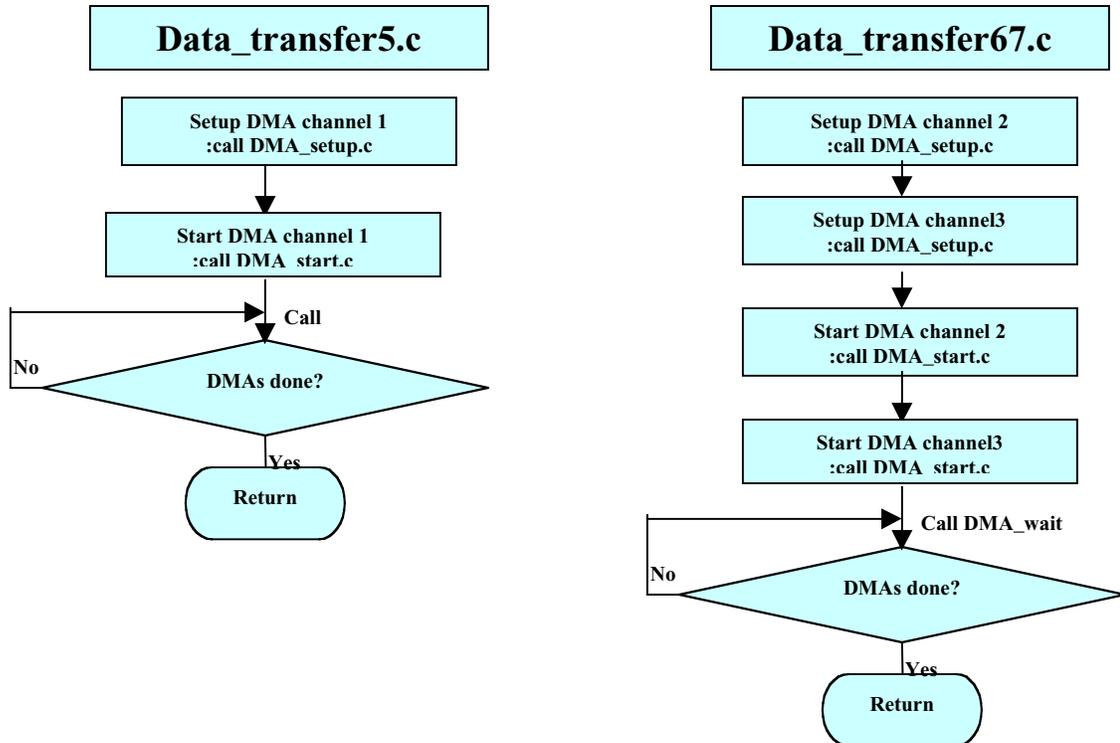


Figure 16. Flowchart of Feedforward-Control-Output Computation (continued)

6.1 Source Modules and Functional Description

The software is mainly written in ANSI C language to achieve a modular and readable software structure. There are three ways to program for DMA services:

1. Use the TI Peripheral Interface Library,
2. Use the TI Chip Support Library, and
3. Use the user-developed programs.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

Peripheral Interface Library and Chip Support Library provide header files that define the memory-mapped DMA controller registers addresses; provide bit maps of each register; and provide macros to define loading, reading, writing, setting, etc. These header files save users from having to perform the tedious jobs. In addition to header files, Chip Support Library provides several modules to open, start, reset, close, and check the status of the DMAs. In our application of DMAs, we used parts of Chip Support Library to develop user programs that minimized software-development time.

```
void DMA_setup(DMA_Handle hDma, unsigned int prctl, unsigned int secctl,  
              unsigned int srcadd, unsigned int desadd,  
              unsigned int xfrctl,  
              unsigned int idxA, unsigned int idxB)
```

This module sets the DMA-controller register values. The calling module defines the DMA channel number, the primary control register value, the secondary control register value, the source address register value, the destination address value, the transfer-counter register value, the global index register A value, and the global index register B value. The data type `DMA_Handle` is defined in the Chip Support Library's header file. The module uses the `DMA_configArgs()` module, provided by Chip Support Library, to load register values.

```
void data_transfer123a4a()
```

This module services DMA #1, DMA #2, DMA #3-a, and DMA #4-a. It opens four DMA channels, loads register values, starts four DMA services, checks completion of DMA services, and closes four DMA channels. The user-defined `DMA_setup()` module is used for loading register values for three DMA channel registers. We used Chip Support Library modules `DMA_open()`, `DMA_start()`, `DMA_wait()`, and `DMA_close()` to develop user programs that open, start, check completion of, and close the DMAs.

```
void data_transfer3b4b()
```

This module services DMA #3-b and DMA #4-b. It opens two DMA channels, loads register values, starts two DMA services, checks completion of DMA services, and closes two DMA channels. The user-defined `DMA_setup()` module is used for loading register values for

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

three DMA channel registers. We used Chip Support Library modules *DMA_open()*, *DMA_start()*, *DMA_wait()* , and *DMA_close()* to develop user programs that open, start, check completion of, and close the DMAs.

```
void data_transfer5()
```

This module services DMA #5. It opens one DMA channel, loads register values, starts one DMA service, checks completion of DMA services, and closes one DMA channel. The user-defined *DMA_setup()* module is used for loading register values for one DMA channel register. We used Chip Support Library modules *DMA_open()*, *DMA_start()*, *DMA_wait()* , and *DMA_close()* to develop user programs that open, start, check completion of, and close the DMAs.

```
void data_transfer67()
```

This module services DMA #6 and DMA #7. It opens two DMA channels, loads register values, starts two DMA services, checks completion of DMA services, and closes two DMA channels. The user-defined *DMA_setup()* module is used for loading register values for two DMA channel registers. We used Chip Support Library modules *DMA_open()*, *DMA_start()*, *DMA_wait()* , and *DMA_close()* to develop user programs that open, start, check completion of, and close the DMAs.

```
void ILC_FFD(int sample_no)
```

This module is the core of module *CSL_DMA0123_and_INTS_ILC()*. Using the transferred data, the module computes the feedforward-control output corresponding to the Pulse ID.¹

¹ Sung-il Kwon, Amy Regan, and Yi-Ming Wang, SNS Superconducting RF Cavity Modeling-Iterative Learning Control, *Nuclear Instruments and Methods in Physics Research Section A*, vol. 482, pp. 12-31, 2002.

7. EXAMPLES

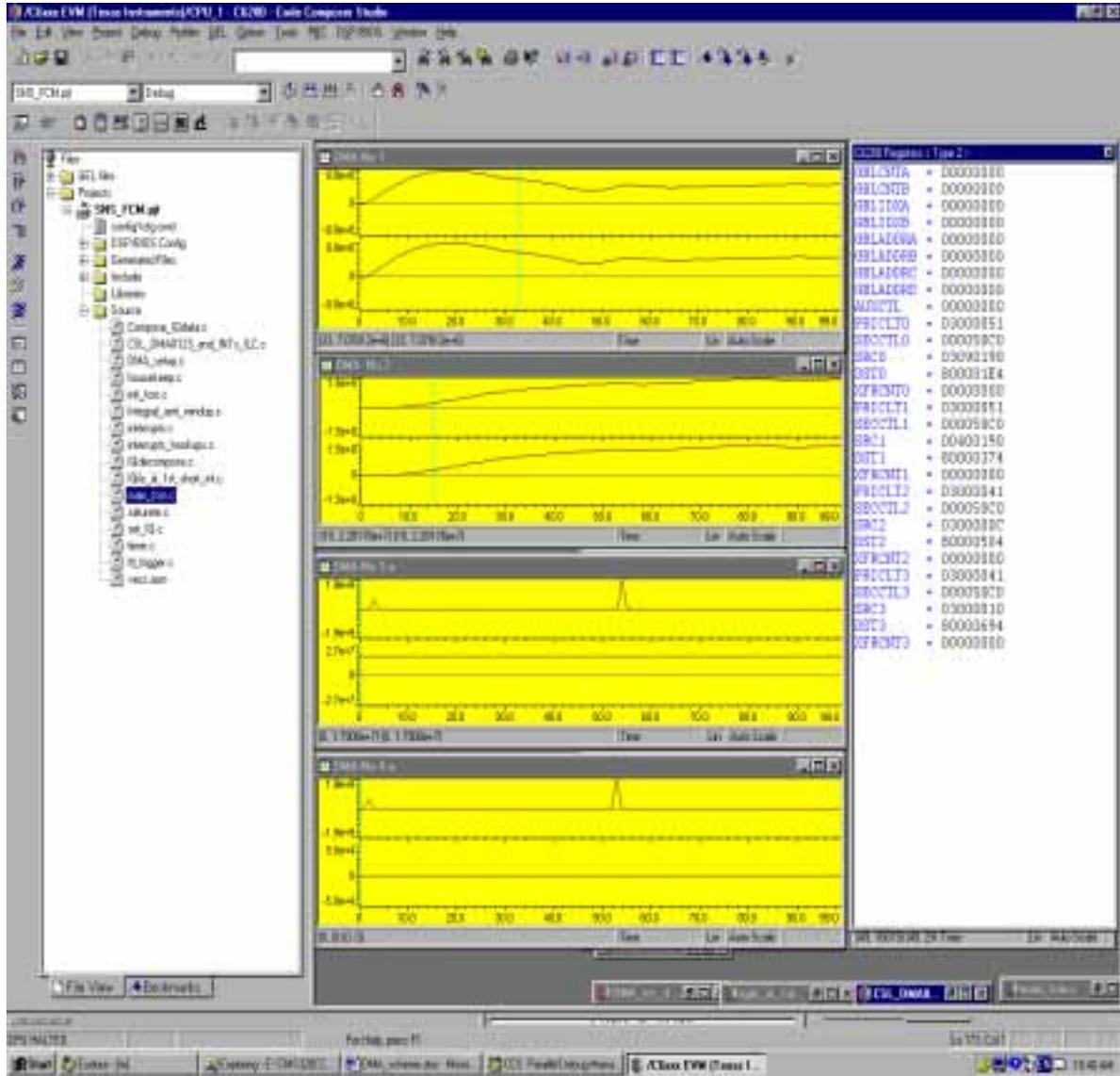


Figure 17 DMA *Data_transfer123a4a0*. One hundred data-sample transfers are considered. The right-half plane numerical data shows the DMA controller registers' data after the DMA data transfers are completed, but while the DMA channels are still open. In the left-half plane plots, DMA #1 (DMA channel 0) and DMA #2 (DMA channel 1) increase the source and destination addresses by 4 bytes. In the cases of DMA No 3-a (DMA channel 2) and DMA #4-a (DMA channel 3), the source addresses do not increase or decrease (sources are FIFOs), but the destination addresses increase by 4 bytes at each data transfer. These are shown in the DMA controller register values in the right-half plane numerical data, where the source addresses for DMA channels 2 and 3 do not change from the values given in Table 5, but the destination addresses increase. The destination memory spaces are filled with the data 17907967 of 0x0300000C and 0 of 0x03000010, respectively, as shown in the destination plots.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

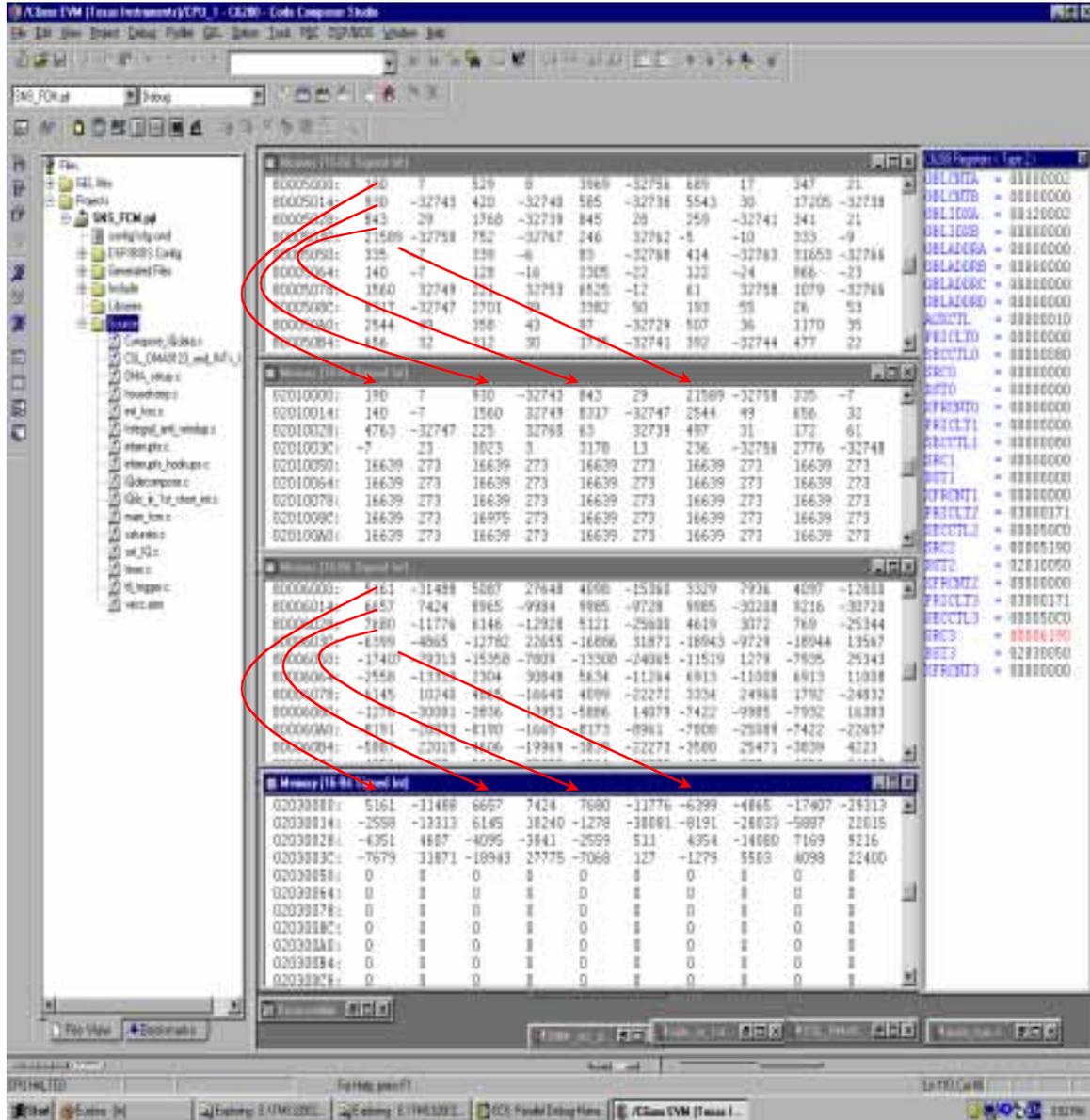


Figure 18 DMA transfers, *data_transfer3b4b0*, where DMA channel 2 and DMA channel 3 are used. In this example, the decimation number is 5 words (4 bytes), and there are 100 data samples. Hence, the frame counter is $100/5=20$ (0x0014) and the element counter is 2 (0x0002). Each frame is composed of 16-bit I data (LSB) and 16-bit Q data (MSB), and hence the Transfer-Counter Registers (XFRcnt2, XFRcnt3) are loaded with 0x00140002. The frame index is $5*4-2=18$ (bytes) (0x0012), and the element index is 2 bytes (0x0002), and hence the Global Index Register A (GBLIDXa) is loaded with 0x00120002. Each Global Index Register (A or B) has its own Global Count Reload Register. For example, Global Index Register A uses Global Index Register A (GBLCNTa). GBLCNTa would be loaded with the element counter value 0x0002. At the end of each frame transfer, the data would be reloaded from GBLCNTa to the element counters of both XFRcnt2 and XFRcnt3. After data transfers are completed, transfer counter registers decrease to zero, as shown in the register values of XFRcnt2, XFRcnt3.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

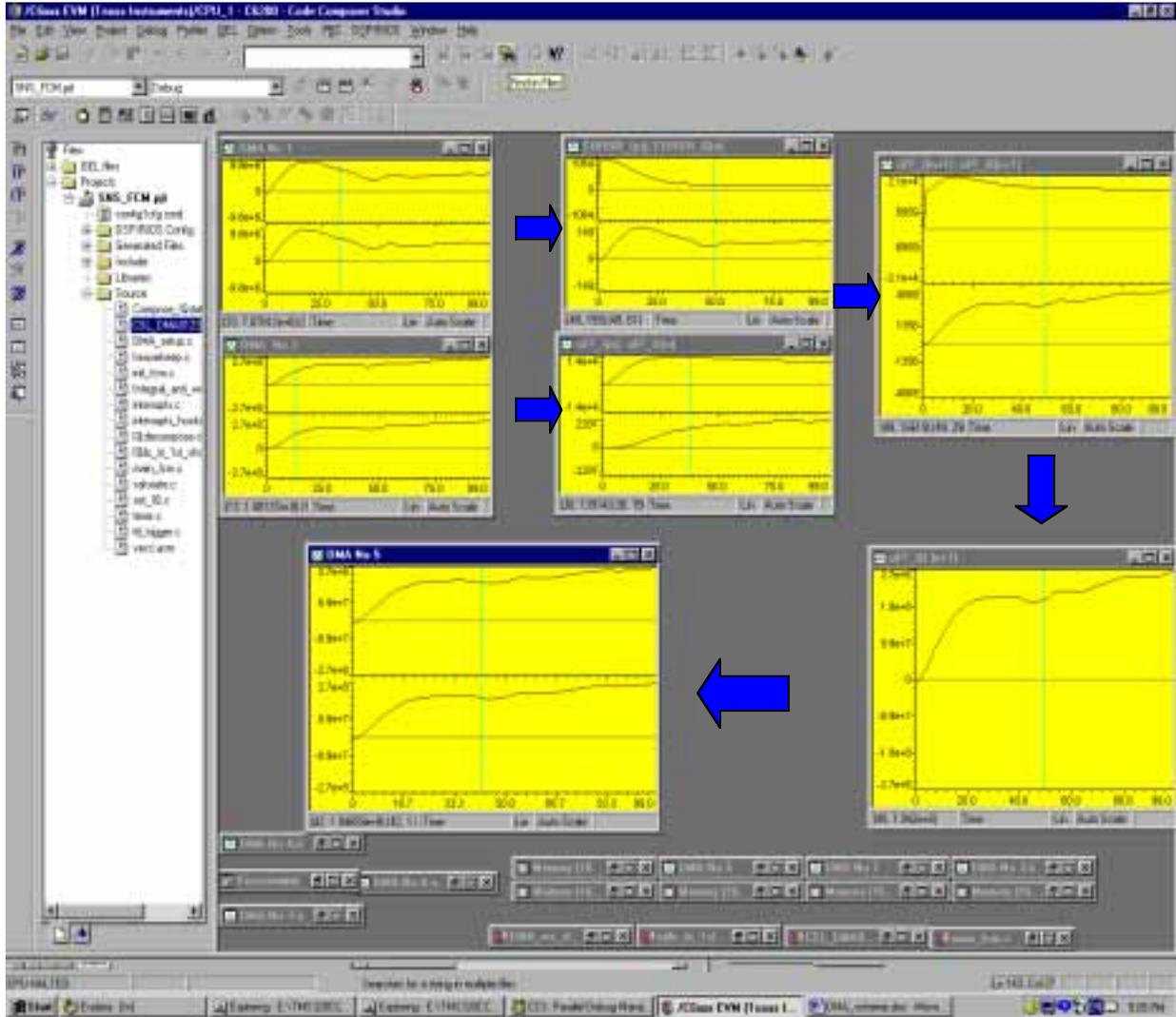


Figure 19 The error data transferred from the dual-port RAM to the DSP internal data memory (DMA #1) and the data transferred from asynchronous SRAM to the DSP's internal data memory (DMA #2) are decomposed into Q15-format error I and error Q ($ERROR_I(n)$, $ERROR_Q(n)$), and Q15-format current-feedforward-control I and current-feedforward-control Q ($uFF_I(n)$, $uFF_Q(n)$) (ILC_FFD0). Based on this decomposed data, the feedforward-control I and feedforward-control Q are updated ($uFF_I(n+1)$, $uFF_Q(n+1)$) (ILC_FFD0). Then, the updated feedforward I and Q are composed into 32-bit data ($uFF_IQ(n+1)$) (ILC_FFD0). When the previous pulse status accompanying the prepulse of the next RF pulse is good, the feedforward-control table values corresponding to the current Pulse ID, residing in asynchronous SRAM, are updated via DMA.

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

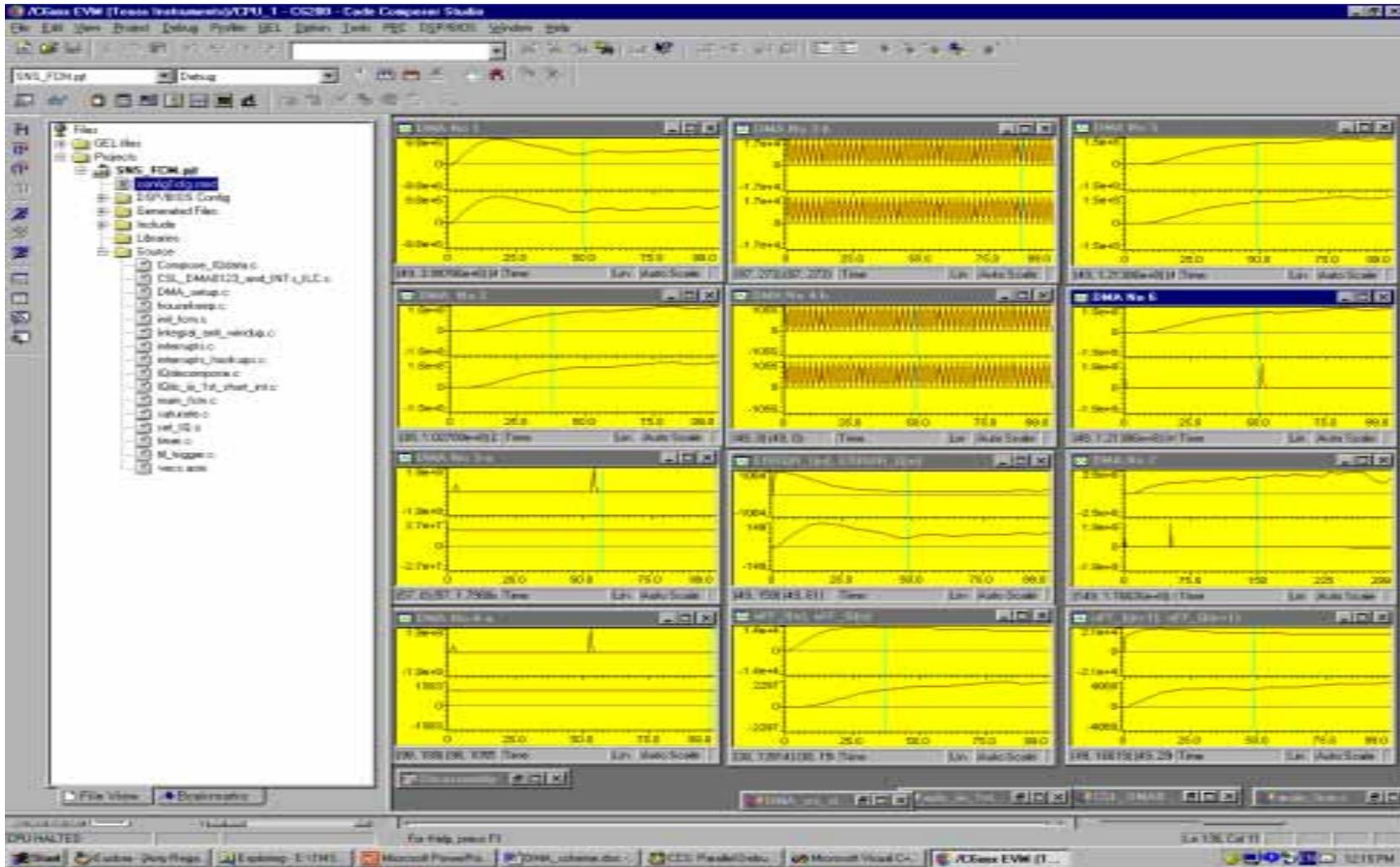


Figure 20. DMA and Feedforward-Control-Output Computation Performed with the TMS320C6201 EVM. The data size is 100 words

Direct Memory Access for the Field- and Resonance-Control Module of the SNS Low-Level RF Control System

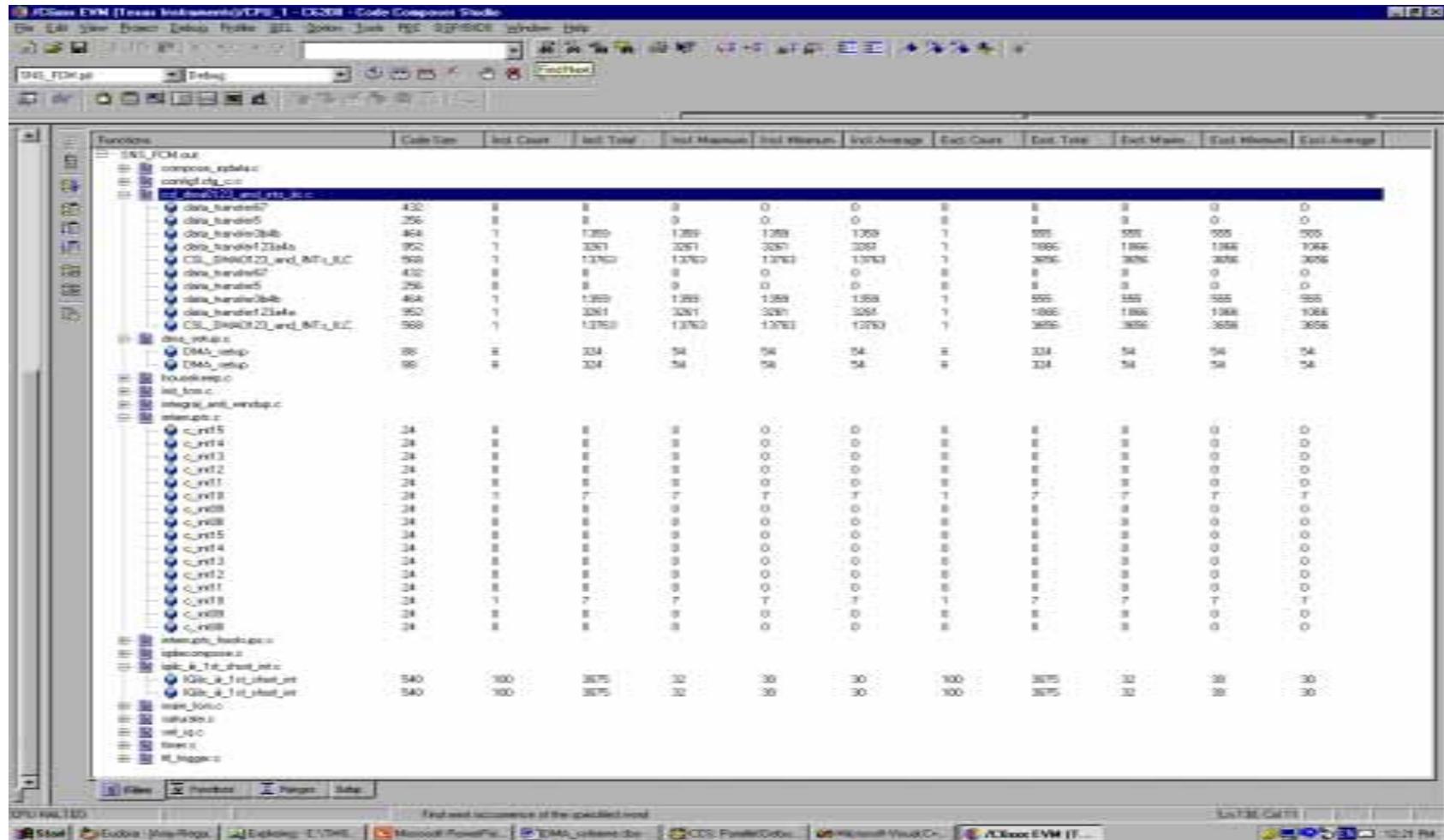


Figure 21. Computational Time Calculation. The computation time is given by the CPU clock cycle. The simulation is performed using a TMS320C6201 EVM. The data size is 100 words. Note that the structure of the DMA in the TMS320C6201 differs from that in the TMS320C6203. We will use the TMS320C6203. The TMS320C6201 has a single FIFO; in contrast, the TMS320C6203 has four FIFOs, one FIFO for each DMA channel.