FEB2 ASIC Configuration

Feb 22, 2023
Updated Nov 29, 2023

The FEB2 slice testboard and prototype board designs include several different ASICs that need to be configured to operate correctly. In the slice testboard design these chips include the COLUTAv3, LAUROC2 and lpGBTv0, while in the prototype board design these chips include the COLUTAv4, ALFE2 and lpGBTv1. These chips are configured using the I2C interface provided by onboard lpGBTs. This document will describe how each type of chip is configured using the lpGBT I2C interface, the default configuration required for each chip and the specific I2C commands required to place each chip in its default configuration.

This document assumes that control lpGBTs registers can be arbitrarily written and read back by the software interface. In the Nevis test setup this register interface is provided by the atlas-tdaq-felix and libflx-lpgbt software packages through the FELIX card optical interface to the control lpGBTs. The I2C interface between an lpGBT and each chip is controlled through the lpGBT I2C control registers as described in section 12 of the lpGBT manual.

COLUTAv4 Configuration

The COLUTA configuration data is written through a control lpGBT I2C bus using 10-bit addressing write commands each containing 8 data bytes. COLUTA configuration bits are divided between global and channel-specific bits as described in the COLUTAv4 datasheet. For channel-specific configuration bits the I2C write command data bytes include two bytes containing subaddressing information and 6 bytes of configuration data. For global configuration bits each I2C write command includes 8 bytes of configuration data. The lpGBT “I2C_WRITE_MULTI_EXT” command is required to send these multiple bytes of configuration data using 10-bit addressing instead of the “I2C_WRITE_MULTI” command used for other ASICs.

12.3.15 I2C_WRITE_MULTI_EXT (0xE)

Writing this command will immediately start a write transaction on the I2C bus. The user must write the correct Address word BEFORE writing this command. On the I2C bus, the master first transmits the 7-bit slave address with R/W=0, and then the Data bytes. The Data bytes are those previously written to the master using the commands I2C_W_MULTI_4BYTE3,2,1,0. The number of transmitted Data bytes is according to the value of bits [6:2] of the Control Register.

Table 12.20: I2C_WRITE_MULTI_EXT command

AddressExt	Bits[9:7] of 10-bit I2C address of target slave
Address	Bits[6:0] of 10-bit I2C address of target slave
Data0	unused
Data1	unused
Data2	unused
Data3	unused
Read	unused
ReadByte	unused

COLUTAv4 I2C Addresses

The I2C addresses of COLUTA chips assembled on different prototype boards are summarised below for reference. These are the base I2C addresses and as will be described below are modified when addressing different ranges of the global configuration register.

On the slice testboard the COLUTAv3 chips have the following 10-bit I2C addresses:

COLUTAv3 Number	**CID Pin Levels [3:0]**	**10-bit I2C Address**
COLUTA13	0101	0101000000
COLUTA14	0110	0110000000
COLUTA15	0111	0111000000
COLUTA16	1000	1000000000
COLUTA17	0001	0001000000
COLUTA18	0010	0010000000
COLUTA19	0011	0011000000
COLUTA20	0100	0100000000

On the FEB2 prototype board v1 the COLUTAv4 chips have the following 10-bit I2C addresses:

COLUTAv3 Number	**CID Pin Levels [3:0]**	**10-bit I2C Address**
COLUTA1	0000	0000000000
COLUTA2	0001	0001000000
COLUTA3	0010	0010000000
COLUTA4	0011	0011000000
COLUTA5	0100	0100000000
COLUTA6	0101	0101000000
COLUTA16	1111	1111000000
COLUTA32	1111	1111000000

COLUTAv4 Channel Configuration Bits

544 channel configuration bits sent as 68 bytes divided between 12 individual I2C write commands. Each I2C write includes 6 config bytes and 2 subaddressing fields defined below. Note there is some duplication in the data written between write #10 and #11.

#	BYTE0	BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	BYTE6	BYTE7
0	ADCSEL	W+R+0x0	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5
1	ADCSEL	W+R+0x3	DATA6	DATA7	DATA8	DATA9	DATA10	DATA11
2	ADCSEL	W+R+0x6	DATA12	DATA13	DATA14	DATA15	DATA16	DATA17
3	ADCSEL	W+R+0x9	DATA18	DATA19	DATA20	DATA21	DATA22	DATA23
4	ADCSEL	W+R+0xC	DATA24	DATA25	DATA26	DATA27	DATA28	DATA29
5	ADCSEL	W+R+0xF	DATA30	DATA31	DATA32	DATA33	DATA34	DATA35
6	ADCSEL	W+R+0x12	DATA36	DATA37	DATA38	DATA39	DATA40	DATA41
7	ADCSEL	W+R+0x15	DATA42	DATA43	DATA44	DATA45	DATA46	DATA47
8	ADCSEL	W+R+0x18	DATA48	DATA49	DATA50	DATA51	DATA52	DATA53
9	ADCSEL	W+R+0x1B	DATA54	DATA55	DATA56	DATA57	DATA58	DATA59
10	ADCSEL	W+R+0x1E	DATA60	DATA61	DATA62	DATA63	DATA64	DATA65
11	ADCSEL	W+R+0x1F	DATA62	DATA63	DATA64	DATA65	DATA66	DATA67

DATAX = channel configuration data byte, number X goes between 0 to 67
ADCSEL = Channel # select, 1-hot encoding (0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80)
W = WRITE FLAG
R = READBACK FLAG
The format of BYTE1 specifically is:
BYTE1[7] : WRITE FLAG
BYTE1[6] : READBACK FLAG
BYTE1[4:0] = SUBADDRESS

The subAddr field specifies which part of the COLUTA channel configuration bit space is being written and can take the values 0,3,6,9,12,15,18,21,24,27,30,31.

Example channel configuration bits for channel 1 of a COLUTAv4 following the table above would be:

Write #	I2C Write Data in Two 4-Byte Groups
0	['0x1', '0x80', '0x4d', '0x18'] ['0x5', '0x82', '0x1', '0x1']
1	['0x1', '0x83', '0x2', '0x8'] ['0x38', '0x80', '0x0', '0x4']

2	['0x1', '0x86', '0x30', '0x80'] ['0x2', '0x40', '0x0', '0x10']
3	['0x1', '0x89', '0x0', '0x7'] ['0x0', '0x0', '0x0', '0x4']
4	['0x1', '0x8c', '0x0', '0x10'] ['0x0', '0x30', '0x0', '0x80']
5	['0x1', '0x8f', '0x0', '0x40'] ['0x1', '0x0', '0x3', '0x0']
6	['0x1', '0x92', '0x7', '0x0'] ['0x0', '0x0', '0x0', '0x0']
7	['0x1', '0x95', '0x0', '0x0'] ['0x68', '0x9a', '0xa6', '0x69']
8	['0x1', '0x98', '0x9a', '0xa6'] ['0xb9', '0xb9', '0xb9', '0x59']
9	['0x1', '0x9b', '0xe0', '0x0'] ['0x0', '0x0', '0x0', '0x0']
10	['0x1', '0x9e', '0x0', '0x0'] ['0x88', '0xa8', '0x1e', '0x9']
11	['0x1', '0x9f', '0x88', '0xa8'] ['0x1e', '0x9', '0x9', '0xc0']

COLUTAv4 Global Configuration Bits

Global COLUTA configuration bits are sent in two I2C writes each containing eight bytes of configuration data. The 10-bit COLUTA I2C address is modified between these two write, adding 0x8 when sending the lower eight-bytes and 0x10 when sending the upper eight bytes. Example global configuration bits are below:

Write #	10-bit Address Modifier	I2C Write Data in Two 4-Byte Groups
0	0x8	['0x14', '0x41', '0x40', '0x4'] ['0xa0', '0xc2', '0x1f', '0x0']
1	0x10	['0x4', '0x84', '0xb9', '0x76'] ['0xd7', '0x8a', '0xd9', '0x7a']

ALFE2 Configuration

The ALFE2 has 16 8-bit registers used to store configuration data and which are defined in the ALFE2 datasheet. The registers are configured using I2C write commands in a three step process, where the 2 LSBs of the ALFE2 7-bit I2C address act as a control register and are modified for each step:
1. Set control register to 0 and send register number via I2C single-byte write command
2. Set control register to 1 and send 0 via I2C single-byte write command
3. Set control register to 2 and register value via I2C single-byte write command

The ALFE2 is typically configured with data suitable for using 25 or 50 Ohm input impedances as defined in the datasheet. Example configuration bit writes for both these cases are summarized below.

ALFE2 25Ohm Input Impedance Configuration

Register #	Configuration Bits

0	0xf0
1	0x3f
2	0xff
3	0xf
4	0xc0
5	0xc7
6	0x1
7	0xfe
8	0x7
9	0x92
10	0xe4
11	0x7f
12	0x0
13	0xc3
14	0x4f
15	0xa7

ALFE2 50Ohm Input Impedance Configuration

Register #	Configuration Bits
0	0x50
1	0xd1
2	0xff
3	0x7
4	0xe0
5	0xc7
6	0x3
7	0xfe
8	0x7
9	0x92
10	0xe4
11	0x7f
12	0x0
13	0xc3
14	0x4f
15	0xa7

lpGBTv0 Configuration

The slice testboard lpGBTs are configured through a register interface. Control lpGBTs are pre-configured with a default working configuration that is updated using the transceiver interface. Data lpGBTs are configured via I2C by a control lpGBT following the procedure described in section 3.5.1 of the lpGBT datasheet. The control lpGBT I2C write command process is defined in section 12.3 of the lpGBT datasheet. The lpGBT register writing procedure is copied here for reference:

1. Master transmits START command.
2. Master transmits the 7-bit lpGBT address followed by the 8th bit (R/W) set to zero.
3. Master transmits bits [7:0] of the register address.
4. Master transmits bits [15:8] of the register address.
5. Master transmits 8-bit register data word (can be repeated).
6. Master transmits STOP command.

After step 5, the register address is automatically incremented. This feature allows a block of consecutive registers to be written in one sequence.

An example list of register values used to configure data lpGBT9 is copied below. In practice multiple consecutive registers are written at the same time using the “I2C_WRITE_MULTI” write command described in section 12.3.13 of the lpGBT manual. A corresponding set of multi-register write values is also copied below as an example.

12.3.13 I2C_WRITE_MULTI (0xC)

Writing this command will immediately start a write transaction on the I2C bus. The user must write the correct Address word BEFORE writing this command. On the I2C bus, the master first transmits the 7-bit slave address with R/W=0, and then the Data bytes. The Data bytes are those previously written to the master using the commands I2C_W_MULTI_4BYTE3,2,1,0. The number of transmitted Data bytes is according to the value of bits [6:2] of the Control Register.

Table 12.18: I2C_WRITE_MULTI command

AddressExt	unused
Address	7-bit I2C address of target slave
Data0	unused
Data1	unused
Data2	unused
Data3	unused
Read	unused
ReadByte	unused

Example data lpGBT9 registers and configuration values

Reg. #	Reg. Value
0x0	0x0
0x1	0x0
0x2	0x0
0x3	0x0
0x4	0x0

0x5	0x0
0x6	0x0
0x7	0x0
0x8	0x0
0x9	0x0
0xa	0x0
0xb	0x0
0xc	0x0
0xd	0x0
0xe	0x0
0xf	0x0
0x10	0x0
0x11	0x0
0x12	0x0
0x13	0x0
0x14	0x0
0x15	0x0
0x16	0x0
0x17	0x0
0x18	0x0
0x19	0x0
0x1a	0x0
0x1b	0x0
0x1c	0x0
0x1d	0x0
0x1e	0x0
0x1f	0x0
0x20	0xc8
0x21	0x38
0x22	0x44
0x23	0x55
0x24	0x55
0x25	0x55
0x26	0x55
0x27	0x55
0x28	0x5
0x29	0x1b

0x2a	0x0
0x2b	0x0
0x2c	0x88
0x2d	0x89
0x2e	0x99
0x2f	0xa
0x30	0xa
0x31	0xa
0x32	0xa
0x33	0x35
0x34	0x51
0x35	0x0
0x36	0x80
0x37	0x0
0x38	0x0
0x39	0x20
0x3a	0x0
0x3b	0x0
0x3c	0x1
0x3d	0x0
0x3e	0x0
0x3f	0x0
0x40	0x0
0x41	0x0
0x42	0x0
0x43	0x0
0x44	0x0
0x45	0x0
0x46	0x0
0x47	0x0
0x48	0x0
0x49	0x0
0x4a	0x0
0x4b	0x0
0x4c	0x0
0x4d	0x0
0x4e	0x0

0x4f	0x0
0x50	0x0
0x51	0x0
0x52	0x0
0x53	0x0
0x54	0x0
0x55	0x0
0x56	0x0
0x57	0x0
0x58	0x0
0x59	0x0
0x5a	0x0
0x5b	0x0
0x5c	0x21
0x5d	0x0
0x5e	0x0
0x5f	0x7d
0x60	0x0
0x61	0xb8
0x62	0x21
0x63	0xa0
0x64	0x0
0x65	0x7d
0x66	0x0
0x67	0xb8
0x68	0x0
0x69	0x0
0x6a	0x0
0x6b	0x0
0x6c	0x61
0x6d	0x0
0x6e	0x0
0x6f	0x0
0x70	0x61
0x71	0x0
0x72	0x0
0x73	0x0

0x74	0x0
0x75	0x0
0x76	0x0
0x77	0x0
0x78	0x0
0x79	0x0
0x7a	0x0
0x7b	0x0
0x7c	0x0
0x7d	0x0
0x7e	0x0
0x7f	0x0
0x80	0x0
0x81	0x0
0x82	0x2d
0x83	0x60
0x84	0x29
0x85	0x60
0x86	0x0
0x87	0x0
0x88	0x0
0x89	0x0
0x8a	0x0
0x8b	0x0
0x8c	0x0
0x8d	0x0
0x8e	0x0
0x8f	0x0
0x90	0x0
0x91	0x0
0x92	0x2d
0x93	0x60
0x94	0x0
0x95	0x0
0x96	0x29
0x97	0x60
0x98	0x29

0x99	0x60
0x9a	0x2c
0x9b	0x60
0x9c	0x0
0x9d	0x0
0x9e	0x0
0x9f	0x0
0xa0	0x0
0xa1	0x0
0xa2	0x0
0xa3	0x0
0xa4	0x0
0xa5	0x0
0xa6	0x0
0xa7	0xa8
0xa8	0x0
0xa9	0x55
0xaa	0x55
0xab	0x3
0xac	0x3
0xad	0x3
0xae	0x3
0xaf	0x3
0xb0	0x3
0xb1	0x3
0xb2	0x3
0xb3	0x3
0xb4	0x3
0xb5	0x3
0xb6	0x3
0xb7	0x3
0xb8	0x3
0xb9	0x3
0xba	0x3
0xbb	0x3
0xbc	0x0
0xbd	0x0

0xbe	0x0
0xbf	0x0
0xc0	0x0
0xc1	0x0
0xc2	0x0
0xc3	0x0
0xc4	0x58
0xc5	0x58
0xc6	0x58
0xc7	0x58
0xc8	0x58
0xc9	0x58
0xca	0x58
0xcb	0x0
0xcc	0x62
0xcd	0x93
0xce	0x62
0xcf	0x93
0xd0	0x72
0xd1	0x93
0xd2	0x72
0xd3	0x93
0xd4	0x82
0xd5	0x93
0xd6	0x82
0xd7	0x93
0xd8	0x82
0xd9	0x93
0xda	0x82
0xdb	0x93
0xdc	0x72
0xdd	0x93
0xde	0x92
0xdf	0x93
0xe0	0x62
0xe1	0x93
0xe2	0x62

0xe3	0x93
0xe4	0x82
0xe5	0x93
0xe6	0x82
0xe7	0x93
0xe8	0x7
0xe9	0x0
0xea	0x0
0xeb	0x0
0xec	0x0
0xed	0x0
0xee	0x0
0xef	0x6

Example data lpGBT 9 multi-register write configuration values

Register #	Consecutive Register Values
0	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
6	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
12	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
18	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
24	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
30	['0x0', '0x0', '0xc8', '0x38', '0x44', '0x55']
36	['0x55', '0x55', '0x55', '0x55', '0x5', '0x1b']
42	['0x0', '0x0', '0x88', '0x89', '0x99', '0xa']
48	['0xa', '0xa', '0xa', '0x35', '0x51', '0x0']
54	['0x80', '0x0', '0x0', '0x20', '0x0', '0x0']
60	['0x1', '0x0', '0x0', '0x0', '0x0', '0x0']
66	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
72	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
78	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
84	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
90	['0x0', '0x0', '0x21', '0x0', '0x0', '0x7d']
96	['0x0', '0xb8', '0x21', '0xa0', '0x0', '0x7d']
102	['0x0', '0xb8', '0x0', '0x0', '0x0', '0x0']
108	['0x61', '0x0', '0x0', '0x0', '0x61', '0x0']
114	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
120	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
126	['0x0', '0x0', '0x0', '0x0', '0x2d', '0x60']

132	['0x29', '0x60', '0x0', '0x0', '0x0', '0x0']
138	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
144	['0x0', '0x0', '0x2d', '0x60', '0x0', '0x0']
150	['0x29', '0x60', '0x29', '0x60', '0x2c', '0x60']
156	['0x0', '0x0', '0x0', '0x0', '0x0', '0x0']
162	['0x0', '0x0', '0x0', '0x0', '0x0', '0xa8']
168	['0x0', '0x55', '0x55', '0x3', '0x3', '0x3']
174	['0x3', '0x3', '0x3', '0x3', '0x3', '0x3']
180	['0x3', '0x3', '0x3', '0x3', '0x3', '0x3']
186	['0x3', '0x3', '0x0', '0x0', '0x0', '0x0']
192	['0x0', '0x0', '0x0', '0x0', '0x58', '0x58']
198	['0x58', '0x58', '0x58', '0x58', '0x58', '0x0']
204	['0x62', '0x93', '0x62', '0x93', '0x72', '0x93']
210	['0x72', '0x93', '0x82', '0x93', '0x82', '0x93']
216	['0x82', '0x93', '0x82', '0x93', '0x72', '0x93']
222	['0x92', '0x93', '0x62', '0x93', '0x62', '0x93']
228	['0x82', '0x93', '0x82', '0x93', '0x7', '0x0']
234	['0x0', '0x0', '0x0', '0x0', '0x0', '0x6']