efabless.com

Caravel SoC Register
Technical Reference Manual (TRM)

Contents

1. Register Mapping 3
   - 1.1. Housekeeping Registers 3
   - 1.2. SoC Management Registers 4
2. Housekeeping Registers 5
   - 2.1. User Project Registers 5
   - 2.2. Housekeeping Registers 13
3. SoC Management Registers 17
   - 3.1. Control 17
   - 3.2. Debug 18
   - 3.3. Flash 18
   - 3.4. General Purpose I/O 19
   - 3.5. LA Input and Output 20
   - 3.6. SPI 23
   - 3.7. Timer 25
   - 3.8. UART 26
   - 3.9. IRQ 28
4. Address Space Mapping 35

1. Register Mapping

Registers are small, fast storage elements within the Caravel SoC that are used for temporary storage of data, intermediate results, and control information during computation. The Register Mapping Technical Reference Manual (TRM) lists the registers in mapping tables in address order for the Caravel System on a Chip (SoC). This TRM is divided into two sections

1.1. Housekeeping Registers

Housekeeping registers are connectd to the housekeeping SPI bus to give access to certain system values and controls independently of the CPU. The housekeeping SPI can be accessed even when the CPU is in full reset. Some control registers in the housekeeping SPI affect the behaviour of the CPU in a way that can be potentially detrimental to the CPU operation, such as adjusting the trim value of the digital frequency-locked loop generating the CPU core clock.

Under normal working conditions, the SPI should not need to be accessed unless it is to adjust the clock speed of the CPU. All other functions are purely for test and debug.

The housekeeping SPI can be accessed by the CPU from a running program by enabling the SPI controller, and enabling the bit that connects the internal SPI controller directly to the housekeeping SPI. This configuration then allows a program to read, for example, the user project ID of the chip. See the SPI controller description for details.

The housekeeping entries in the Housekeeping Register Section follow the format shown below.

register_name																															
0x1234567B								0x1234567A								0x12345679								0x12345678							
								0x01								0x02								0x03							
(undefined, reads zero)								Register field contents																Register field contents							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 0-23 Bit field decription

Register **register_name** description

The first line contains the register name which corresponds to the macro definition in defs.h which can be used to access the register memory from a C program running on flash. The second line shows the the (byte) hex addresses of the 32-bit register. Normally the register is read as a 32-bit word using the lowest address. Each register can be reached from the housekeeping SPI interface. The SPI address of

each register is 7 bits and is independent of the memory-mapped address. The SPI address for each byte is shown on line three. The fourth line shows which bit fields of the register contain relevant data, and groups them into named sections. The fifth line gives the bit values, for reference. Under the main table is a description of any bit fields declared on the sixth line of the table above, including a default value and any relevant notes. The last lines for each register contain a general description of the register and its use.

1.2. SoC Management Registers

In the context of SoCs, management registers serve as a bridge between hardware and software components. These registers allow software (such as device drivers) to interact with specific hardware functionalities. Register management is essential for ensuring proper communication between software running on the CPU and the underlying hardware components. The management entries in the Management Register section are shown as follows.

register_name																															
0x1234567B								0x1234567A								0x12345679								0x12345678							
(undefined, reads zero)								Register field contents																Register field contents							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 0-23 Bit field decription

Register **register_name** description

The first line contains the register name which corresponds to the macro definition in defs.h which can be used to access the register memory from a C program running on flash. The second line shows the the (byte) hex addresses of the 32-bit register. Normally the register is read as a 32-bit word using the lowest address. The third line shows which bit fields of the register contain relevant data, and groups them into named sections. The fourth line gives the bit values, for reference. Under the main table is a description of any bit fields declared on the fifth line of the table above, including a default value and any relevant notes. The last lines for each register contain a general description of the register and its use.

2. Housekeeping Registers

This section documents the Housekeeping registers. It lists all the registers in tables in address order.

2.1. User Project Registers

reg_mprj_xfer																													
0x26000003								0x26000002								0x26000001								0x26000000					
																								0x13					
(undefined, reads zero)																								gpio xfer controls					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio xfer controls (7 bits) bit fields:

bit 0	serial xfer/busy	Write 1 to apply configuration values to GPIO. Auto-zeroing. Read back value 1 = busy, 0 = idle
bit 1	bitbang enable	1 = serial transfer bitbang mode enabled; 0 = bitbang mode disabled
bit 2	bitbang resetn	0 = bit bang mode reset; 1 = bitbang mode normal operation
bit 3	bitbang load	0 = bit bang mode normal operation; 1 = latch configuration values
bit 4	bitbang clock	0->1 transition: Advance data in serial shift register by 1 bit in bitbang mode
bit 5	bitbang data right	Value = data to apply to serial data right side shift register (GPIO 0 to 18) on next bitbang clock
bit 6	bitbang data left	Value = data to apply to serial data left side shift register (GPIO 19 to 37) on next bitbang clock

The **reg_mprj_xfer** register controls the programming of the configurable GPIO (general-purpose input/output) pins . There are 38 GPIO pins, in two banks of 19. GPIO 0 to 18 start at the bottom right corner of the chip and extend to the midpoint of the top side of the chip. GPIO 19 to 37 start at the midpoint of the top side of the chip and extend to the bottom left corner. The GPIO indexes always increase in a counterclockwise direction around the chip perimeter. The GPIOs are configured through the "user_defines.v" file to have a specific configuration on power-up. However, the GPIOs may be reprogrammed at any time either through the housekeeping SPI or through processor reads and writes to the memory map described on this page. The programming starts with the intended configuration for each GPIO programmed into registers **reg_mprj_io_0** to **reg_mprj_io_37** (see below). These registers are a *staging area* for the GPIO configuration. The actual GPIO configuration is kept in duplicate registers next to each GPIO and are not directly readable or writeable. The GPIOs are configured by an automatic programming step that copies the configurations from the registers to the GPIO pins via a serial shift register. The **reg_mprj_xfer** register controls this programming process. Normally, only the simple, automatic programming is used. The user sets (through the housekeeping SPI or from a running program) bit 0 to start the configuration transfer from housekeeping registers to the GPIO pins. The same bit 0 may be subsequently monitored; when the bit clears, the configuration programming is complete. The rest of the control bits are intended only for diagnostic purposes. They can be used to override the automatic GPIO programming and perform the programming sequence manually.

reg_mprj_pwr																													
0x26000007								0x26000006								0x26000005								0x26000004					
																								0x6E					
(undefined, reads zero)																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register intended for future use powering up/down internal power domains. Currently it has no function.

reg_mprj_datal																													
0x2600000F								0x2600000E								0x2600000D								0x2600000C					
0x6A								0x6B								0x6C								0x6D					
gpio data [31:0]																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 31 to 0	gpio data [31:0]	Read values from or write values to the GPIO pins for GPIO[31] (bit 31) down to GPIO[0] (bit 0

reg_mprj_datah																															
0x26000013								0x26000012								0x26000011								0x26000010							
																								0x69							
(undefined, reads zero)																								gpio data [37:32]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 5 to 0 gpio data [37:32] Read values from or write values to the GPIO pins for GPIO[37] (bit 5) down to GPIO[32] (bit 0

The **reg_mprj_datah** and **reg_mprj_datal** registers together comprise the 38 bits of data corresponding to the 38 configurable GPIO pins. Writing to the register is only meaningful for GPIO pins that are configured for management control and which are not being controlled by one of the special functions (e.g., SPI master, UART, housekeeping SPI). If the corresponding GPIO is configured for management control, configured as an output, and has the output enabled, then a bit written to these registers will appear as an output value on the corresponding GPIO pin. Reading the value from these registers will read the value at the pin, regardless of whether or not the pin is configured for management control, and whether or not the pin is configured for a special function. However, the GPIO must have the input enabled.

reg_mprj_io_0																															
0x26000027								0x26000026								0x26000025								0x26000024							
																0x1D								0x1E							
(undefined, reads zero)																			gpio configuration												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_1																															
0x2600002B								0x2600002A								0x26000029								0x26000028							
																0x1F								0x20							
(undefined, reads zero)																			gpio configuration												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_2																															
0x2600002F								0x2600002E								0x2600002D								0x2600002C							
																0x21								0x22							
(undefined, reads zero)																			gpio configuration												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_3																															
0x26000033								0x26000032								0x26000031								0x26000030							
																0x23								0x24							
(undefined, reads zero)																			gpio configuration												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_4			
0x26000037	0x26000036	0x26000035	0x26000034
		0x25	0x26
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_5			
0x2600003B	0x2600003A	0x26000039	0x26000038
		0x27	0x28
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_6			
0x2600003F	0x2600003E	0x2600003D	0x2600003C
		0x29	0x2A
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_7			
0x26000043	0x26000042	0x26000041	0x26000040
		0x2B	0x2C
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_8			
0x26000047	0x26000046	0x26000045	0x26000044
		0x2D	0x2E
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_9			
0x2600004B	0x2600004A	0x26000049	0x26000048
		0x2F	0x30
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_10																															
0x2600004F								0x2600004E								0x2600004D								0x2600004C							
																0x31								0x32							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_11																															
0x26000053								0x26000052								0x26000051								0x26000050							
																0x33								0x34							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_12																															
0x26000057								0x26000056								0x26000055								0x26000054							
																0x35								0x36							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_13																															
0x2600005B								0x2600005A								0x26000059								0x26000058							
																0x37								0x38							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_14																															
0x2600005F								0x2600005E								0x2600005D								0x2600005C							
																0x39								0x3A							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_15																															
0x26000063								0x26000062								0x26000061								0x26000060							
																0x3B								0x3C							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_16			
0x26000067	0x26000066	0x26000065	0x26000064
		0x3D	0x3E
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_17			
0x2600006B	0x2600006A	0x26000069	0x26000068
		0x3F	0x40
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_18			
0x2600006F	0x2600006E	0x2600006D	0x2600006C
		0x41	0x42
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_19			
0x26000073	0x26000072	0x26000071	0x26000070
		0x43	0x44
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_20			
0x26000077	0x26000076	0x26000075	0x26000074
		0x45	0x46
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_21			
0x2600007B	0x2600007A	0x26000079	0x26000078
		0x47	0x48
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			gpio configuration												

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_22																															
0x2600007F								0x2600007E								0x2600007D								0x2600007C							
																0x49								0x4A							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_23																															
0x26000083								0x26000082								0x26000081								0x26000080							
																0x4B								0x4C							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_24																															
0x26000087								0x26000086								0x26000085								0x26000084							
																0x4D								0x4E							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_25																															
0x2600008B								0x2600008A								0x26000089								0x26000088							
																0x4F								0x50							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_26																															
0x2600008F								0x2600008E								0x2600008D								0x2600008C							
																0x51								0x52							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_27																															
0x26000093								0x26000092								0x26000091								0x26000090							
																0x53								0x54							
(undefined, reads zero)																				gpio configuration											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_28			
0x26000097	0x26000096	0x26000095	0x26000094
		0x55	0x56
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_29			
0x2600009B	0x2600009A	0x26000099	0x26000098
		0x57	0x58
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_30			
0x2600009F	0x2600009E	0x2600009D	0x2600009C
		0x59	0x5A
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_31			
0x260000A3	0x260000A2	0x260000A1	0x260000A0
		0x5B	0x5C
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_32			
0x260000A7	0x260000A6	0x260000A5	0x260000A4
		0x5D	0x5E
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_33			
0x260000AB	0x260000AA	0x260000A9	0x260000A8
		0x5F	0x60
(undefined, reads zero)		gpio configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_34																													
0x260000AF								0x260000AE								0x260000AD								0x260000AC					
																0x61								0x62					
(undefined, reads zero)																				gpio configuration									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_35																													
0x260000B3								0x260000B2								0x260000B1								0x260000B0					
																0x63								0x64					
(undefined, reads zero)																			gpio configuration										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_36																													
0x260000B7								0x260000B6								0x260000B5								0x260000B4					
																0x65								0x66					
(undefined, reads zero)																			gpio configuration										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields: See bit field definitions for reg_mprj_io_37.

reg_mprj_io_37																													
0x260000BB								0x260000BA								0x260000B9								0x260000B8					
																0x67								0x68					
(undefined, reads zero)																			gpio configuration										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

gpio configuration (13 bits) bit fields:

bit 0	management enable	1 = management SoC controls GPIO; 0 = user project controls GPIO
bit 1	output disable	1 = digital output driver disabled ; 0 = digital output driver enabled (management controlled mode only)
bit 2	hold state value	Value of GPIO when in low-power state.
bit 3	input disable	1 = digital input driver disabled; 1 = digital input driver enabled
bit 4	IB mode select	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bit 5	analog enable	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bit 6	analog select	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bit 7	analog polarity	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bit 8	slow slew	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bit 9	trip point select	See definition in SkyWater documentation for sky130_fd_io__top_gpiov2
bits 10-12	digital mode	See table below for typical settings; see SkyWater documentation for complete list.
000	disabled	Both input and output digital buffers disabled. Use this when connecting an analog signal to the pad
001	input	Digital input only. Output buffer is disabled.
010	input pullup	Input mode with pull-up. User mode only: Output must be enabled and driven to value 1.
011	input pulldown	Input mode with pull-down. User mode only: Output must be enabled and driven to value 0.
110	output	Digital output. User mode only: Output must be enabled (OEB = 0)

The registers **reg_mprj_io_0** through **reg_mprj_io_37** are a staging area for the configuration of GPIO pins 0 to 37, respectively. Each GPIO is controlled by a 13 bit vector with values as described above. The values in the staging area are copied to the GPIO pins by use of the register **reg_mrpj_xfer** (see above). The default value of all GPIOs other than 0 to 4 is determined by the values set by the user in the file **user_defines.v**. The configuration is via-programmed at the GPIO itself. However, the value is not duplicated in the housekeeping registers, which continue to read the default values indicated above until the register is modified. The header file **defs.h** includes macros for the most common GPIO configuration settings.

2.2. Housekeeping Registers

reg_hkspi_status																													
0x26100003								0x26100002								0x26100001								0x26100000					
																								0x00					
(undefined, reads zero)																								SPI status					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 0-7 SPI status This byte is currently unused and undefined (intended for SPI mode control)

reg_hkspi_chip_id																													
0x26100007								0x26100006								0x26100005								0x26100004					
								0x01								0x02								0x03					
(undefined, reads zero)								Manufacturer ID (= 0x456)																Product ID = (0x11)					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 0-23 Fixed product ID value 0x11 representing Caravel v4 (Sky130). Value is read-only
bits 16-23 Fixed manufacturer ID value 0x456 representing Efabless Corporation. Value is read-only

reg_hkspi_user_id																													
0x2610000B								0x2610000A								0x26100009								0x26100008					
0x04								0x05								0x06								0x07					
User project ID																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 0-31 Unique project ID assigned to each user design. Traditionally, bits 16-31 are an ID sub-field for the MPW run.

The **reg_hkspi_user_id** register allows a user to read back the 32-bit project ID value assigned to the Caravel user project. Each project on a reticle (currently 40 projects per shuttle run) gets a unique identification number that is automatically via-programmed into this register during chip assembly prior to tape-out.

reg_hkspi_pll_ena																													
0x2610000F								0x2610000E								0x2610000D								0x2610000C					
																								0x08					
(undefined, reads zero)																												DCO	ENA
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0 ENA DLL/DCO enable (1 = enabled; 0 = disabled)
bit 1 DCO DCO mode enable (1 = use DCO mode; 0 = use DLL mode)

The caravel chip has an on-board oscillator that is a programmable-length ring oscillator, or DCO (digitally-controlled oscillator). The oscillator on the Sky130 version of Caravel has a frequency range of about 50MHz to 120MHz (at a low-voltage supply of 1.8V). This register controls the enabled state of the DCO and the DLL (digital locked loop), which is a controller that automatically trims the DCO to lock the frequency to a multiple of the external clock input. The base enable (bit 0) must be set to 1 for either the DCO or the DLL to operate. The DCO enable bit is effectively a DLL disable bit, which lets the DCO run freely with manual trim control through the **reg_hkspi_pll_trim** register. Registers affecting DCO and DLL operation are **reg_hkspi_pll_ena**, **reg_hkspi_pll_bypass**, **reg_hkspi_pll_trim**, **reg_hkspi_pll_source**, and **reg_hkspi_pll_divider**. Please see the register descriptions below for additional information.

reg_hkspi_pll_bypass			
0x26100013	0x26100012	0x26100011	0x26100010
			0x09
(undefined, reads zero)			BYP

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0 BYP DLL bypass (1 = use external clock; 0 = use DLL/DCO clock)

The **reg_hkspi_pll_bypass** register contains one bit which, when set, bypasses the DLL or DCO (see above) and allows the Caravel chip core to be driven by the signal on the external clock pin. When cleared, the Caravel chip core is driven from the DLL/DCO clock. The default value is 1 so that the clock rate on power-up is defined by the external clock.

reg_hkspi_irq			
0x26100017	0x26100016	0x26100015	0x26100014
			0x0A
(undefined, reads zero)			IRQ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0 IRQ Manual interrupt (1 = interrupt; 0 = no action)

The **reg_hkspi_irq** register can be used to apply a manual interrupt to the VexRISC processor. The value of bit 0 is passed directly to the interrupt vector of the processor.

reg_hkspi_reset			
0x2610001B	0x2610001A	0x26100019	0x26100018
			0x0B
(undefined, reads zero)			RST

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0 RST Manual reset (1 = apply reset; 0 = release reset)

The **reg_hkspi_reset** register can be used to apply a manual reset to the VexRISC processor. It is equivalent to grounding the RESETB pin on the chip. Note that both the pin and software resets do not affect values in the housekeeping module, which can only be reset by a power cycle to the chip.

reg_hkspi_pll_trim			
0x2610001F	0x2610001E	0x2610001D	0x2610001C
0x10	0x0F	0x0E	0x0D
(undefined, reads zero)	DLL manual trim		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

all bits All 1 values = maximum trim; all 0 values = minimum trim (To-do: table needed)

The **reg_hkspi_pll_trim** value controls the frequency of the DCO when the DCO is not in DLL mode (free-running DCO mode). The trim is undecoded, so each trim bit will turn a section of the ring oscillator on or off, lengthening or shortening the ring oscillator. With 26 trim bits, there are effectively 27 unique settings of the DCO, from minimum trim (value 0x00000000) to maximum trim (value 0x03ffffff). The default state is one less than the maximum trim and is necessary for proper startup of the DLL.

reg_hkspi_pll_source																															
0x26100023								0x26100022								0x26100021								0x26100020							
																								0x11							
(undefined, reads zero)																								out div 2			out div 1				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 2-0	out div 1	Core clock DLL output divider (0 = thru; 1 = off; 2-7 = divide by 2 to 7
bits 5-3	out div 2	Secondary (user) clock DLL output divider (0 = thru; 1 = off; 2-7 = divide by 2 to 7

The **reg_hkspi_pll_source** register defines the values of the two output dividers associated with the DCO/DLL. The primary divider (div 1) divides the DCO/DLL frequency down by the given value to generate the primary (core) clock for the processor. The secondary divider (div 2) divides the DCO/DLL frequency down by the given value to generate the secondary (user) clock, which is passed directly to the user project as an independent clock that is independent of the core clock (wishbone clock). Note that value 0 is effectively divide-by-1 (pass-through), while value 1 disables the divider. The output dividers operate in both DCO and DLL modes.

reg_hkspi_pll_divider																															
0x26100027								0x26100026								0x26100025								0x26100024							
																								0x12							
(undefined, reads zero)																											feedback div				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bits 4-0	feedback div	DLL feedback divider. Values 1-31 = divide by 1 to 31

The **reg_hkspi_pll_divider** register defines the value of the feedback divider for the DLL. This divider value is only relevant when the DLL mode is on (DCO mode off). The DLL will lock to a frequency that is the frequency of the external input clock multiplied by the feedback divider value. The feedback divider value must be set such that the resulting DLL frequency is in the operational range of the DCO (which is 50MHz to 120MHz at 1.8V vccd supply). The value of the resulting clock when running in DLL mode can be computed as the external clock frequency times the feedback divider value, divided by the output divider value (see **reg_hkspi_pll_source**).

reserved																															
0x2610002B								0x2610002A								0x26100029								0x26100028							
																								0x1A							
(undefined, reads zero)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved

reg_power_good																													
0x26200003								0x26200002								0x26200001								0x26200000					
																								0x1A					
(undefined, reads zero)																												power monitors	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

power monitors (4 bits) bit fields:

bit 0	vdda2 power good	1 = vdda2 (user 2 3.3V domain) powered up; 0 = vdda2 powered down
bit 1	vdda1 power good	1 = vdda1 (user 1 3.3V domain) powered up; 0 = vdda1 powered down
bit 2	vccd2 power good	1 = vccd2 (user 2 1.8V domain) powered up; 0 = vccd2 powered down
bit 3	vccd1 power good	1 = vccd1 (user 1 1.8V domain) powered up; 0 = vccd1 powered down

The **reg_power_good** read-only bits report the status of the power supply to the four independent user area power supplies. Note that these are trivially simple power detectors and will read "1" whenever a supply is high enough to trip a digital buffer input, which is half of the nominal power supply or less.

reg_clk_out_dest																													
0x26200007								0x26200006								0x26200005								0x26200004					
																								0x1B					
(undefined, reads zero)																												monitors	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

monitors bit fields:

bit 0	trap monitor	1 = Moniitor CPU trap state on GPIO[13] (unavailable on current caravel version)
bit 1	core clock monitor	1 = Monitor core clock on GPIO[14]
bit 2	user clock monitor	1 = Monitor secondary (user) clock on GPIO[15]

The **reg_clk_out_dest** monitoring functions allow either or both of the core clock and the user clock to be routed to a GPIO pin for monitoring. Specific GPIO pins (14 and 15) are assigned to these functions.

reg_irq_source																													
0x2620000F								0x2620000E								0x2620000D								0x2620000C					
																								0x1C					
(undefined, reads zero)																												source	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

source bit fields:

bit 0	IRQ 1 source	1 = Enable pin GPIO[7] as IRQ[1] input source
bit 1	IRQ 2 source	1 = Enable pin GPIO[12] as IRQ[2] input source

The **reg_irq_source** register controls the enablement of special-purpose GPIO pins (7 and/or 12) to be used for external interrupts to the processor from an off-chip source.

reg_hkspi_disable																															
0x26200013								0x26200012								0x26200011								0x26200010							
																								0x6F							
(undefined, reads zero)																															DIS
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0 DIS Disable the housekeeping SPI. 1 = SPI disabled; 0 = SPI enabled

The **reg_hkspi_disable** register contains a single bit that will disable the special function of the housekeeping SPI. The housekeeping SPI should be disabled whenever GPIO pins 1 to 4 are used for any purpose other than the housekeeping SPI interface. Otherwise, the CSB pin (GPIO 3) can trigger an action on the SPI, which may then attempt to apply an output value to SDO (GPIO 1).

3. SoC Management Registers

3.1. Control

CTRL_RESET																															
0xF0000003								0xF0000002								0xF0000001								0xF0000000							
(undefined, reads zero)																														PU_R	OC_R
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	Description
[0]	SOC_RST	Write 1 to this register to reset the full SoC (Pulse Reset)
[1]	CPU_RST	Write 1 to this register to reset the CPU(s) of the SoC (Hold Reset)

CTRL_SCRATCH																															
0xF0000007								0xF0000006								0xF0000005								0xF0000004							
scratch space																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Use this register as a scratch space to verify that software read/write accesses to the Wishbone/CSR bus are working correctly. The initial reset value of 0x1234578 can be used to verify endianness.

CTRL_BUS_ERRORS																															
0xF000000B								0xF000000A								0xF0000009								0xF0000008							
bus errors																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Total number of Wishbone bus errors (timeouts) since start.

3.2. Debug

DEBUG_MODE_OUT			
0xF0000803	0xF0000802	0xF0000801	0xF0000800
GPIO Output Control			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO Output(s) Control. *																															

DEBUG_OEB_OUT			
0xF0001003	0xF0001002	0xF0001001	0xF0001000
GPIO Output Control			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO Output(s) Control. *																															

3.3. Flash

FLASH_CORE_MMAP_DUMMY_BITS			
0xF0001803	0xF0001802	0xF0001801	0xF0001800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

FLASH_CORE_MASTER_CS			
0xF0001807	0xF0001806	0xF0001805	0xF0001804
Flash Enable			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
It acts as an enable for flash if you want to enable the flashing you would write 1 to it that would asset the cs value by making it low																															

FLASH_CORE_MASTER_PHYCONFIG			
0xF000180B	0xF000180A	0xF0001809	0xF0001808
	MASK	WIDTH	LEN

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	Description
[7:0]	LEN	SPI Xfer length (in bits).
[11:8]	WIDTH	SPI Xfer width (1/2/4/8).
[23:16]	MASK	SPI DQ output enable mask (set bits to 1 to enable output drivers on DQ lines).

FLASH_CORE_MASTER_RXTX			
0xF000180F	0xF000180E	0xF000180D	0xF000180C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved

FLASH_CORE_MASTER_STATUS			
0xF0001813	0xF0001812	0xF0001811	0xF0001810

(undefined, reads zero)																														_REA	_REA
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	Description
[0]	TX_READY	TX FIFO is not full.
[1]	RX_READY	RX FIFO is not empty.

FLASH_PHY_CLK_DIVISOR			
0xF0002003	0xF0002002	0xF0002001	0xF0002000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

It is the ratio between the core clock and flashing clock for example if FLASH_PHY_CLK_DIVISOR is 0 the flashing clock would be half the core clock if it's 1 the flashing clock will be 1/4 of the core clock

3.4. General Purpose I/O

GPIO_MODE1			
0xF0002803	0xF0002802	0xF0002801	0xF0002800
GPIO Tristate(s) Control.			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Tristate(s) Control. *

GPIO_MODE0			
0xF0002807	0xF0002806	0xF0002805	0xF0002804
GPIO Tristate(s) Control.			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Tristate(s) Control. *

GPIO_IEN			
0xF000280B	0xF000280A	0xF0002809	0xF0002808
GPIO Tristate(s) Control.			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Tristate(s) Control. *

GPIO_OE																													
0xF000280F								0xF000280E								0xF000280D								0xF000280C					
GPIO Tristate(s) Control.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Tristate(s) Control. *

GPIO_IN																													
0xF0002813								0xF0002812								0xF0002811								0xF0002810					
GPIO Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

GPIO_OUT																													
0xF0002817								0xF0002816								0xF0002815								0xF0002814					
GPIO Ouptut(s) Control.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Ouptut(s) Control. *

3.5. LA Input and Output

LA_IEN3																													
0xF0003003								0xF0003002								0xF0003001								0xF0003000					
LA Input Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 96-127 of LA_IEN. LA Input Enable

LA_IEN2																													
0xF0003007								0xF0003006								0xF0003005								0xF0003004					
LA Input Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 64-95 of LA_IEN.

LA_IEN1																													
0xF000300B								0xF000300A								0xF0003009								0xF0003008					
LA Input Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 32-63 of LA_IEN.

LA_IEN0																													
0xF000300F								0xF000300E								0xF000300D								0xF000300C					
LA Input Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-31 of LA_IEN.

LA_OE3																													
0xF0003013								0xF0003012								0xF0003011								0xF0003010					
LA Output Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 96-127 of LA_OE. LA Output Enable

LA_OE2																													
0xF0003017								0xF0003016								0xF0003015								0xF0003014					
LA Output Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 64-95 of LA_OE.

LA_OE1																													
0xF000301B								0xF000301A								0xF0003019								0xF0003018					
LA Output Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 32-63 of LA_OE.

LA_OE0																													
0xF000301F								0xF000301E								0xF000301D								0xF000301C					
LA Output Enable																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-31 of LA_OE.

LA_IN3																													
0xF0003023								0xF0003022								0xF0003021								0xF0003020					
LA Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 96-127 of LA_IN. LA Input(s) Status.

LA_IN2																													
0xF0003027								0xF0003026								0xF0003025								0xF0003024					
LA Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 64-95 of LA_IN.

LA_IN1																													
0xF000302B								0xF000302A								0xF0003029								0xF0003028					
LA Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 32-63 of LA_IN.

LA_IN0																													
0xF000302F								0xF000302E								0xF000302D								0xF000302C					
LA Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-31 of LA_IN.

LA_OUT3																													
0xF0003033								0xF0003032								0xF0003031								0xF0003030					
LA Ouptut(s) Control.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 96-127 of LA_OUT. LA Ouptut(s) Control.

LA_OUT2																													
0xF0003037								0xF0003036								0xF0003035								0xF0003034					
LA Ouptut(s) Control.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 64-95 of LA_OUT.

LA_OUT1																													
0xF000303B								0xF000303A								0xF0003039								0xF0003038					
LA Ouptut(s) Control.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 32-63 of LA_OUT.

LA_OUT0																															
0xF000303F								0xF000303E								0xF000303D								0xF000303C							
LA Ouptut(s) Control.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-31 of LA_OUT.

MPRJ_WB_IENA_OUT																															
0xF0003803								0xF0003802								0xF0003801								0xF0003800							
GPIO Output(s) Control.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Output(s) Control. *

3.6. SPI

SPI_ENABLED_OUT																															
0xF0004003								0xF0004002								0xF0004001								0xF0004000							
GPIO Output(s) Control.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Output(s) Control. *

SPI_MASTER_CONTROL																															
0xF0004803								0xF0004802								0xF0004801								0xF0004800							
																LENGTH															START
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	Description
[0]	START	Write 1 to this register to reset the full SoC (Pulse Reset)
[15:8]	LENGTH	Write 1 to this register to reset the CPU(s) of the SoC (Hold Reset)

SPI_MASTER_STATUS																															
0xF0004807								0xF0004806								0xF0004805								0xF0004804							
(undefined, reads zero)																															DONE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	Description
[0]	DONE	SPI Xfer Done (when read as 1).

SPI_MASTER_MOSI																													
0xF000480B								0xF000480A								0xF0004809								0xF0004808					
SPI MOSI data																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI MOSI data (MSB-first serialization).

SPI_MASTER_MISO																													
0xF000480F								0xF000480E								0xF000480D								0xF000480C					
SPI MISO data																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI MISO data (MSB-first de-serialization).

SPI_MASTER_CS																													
0xF0004813								0xF0004812								0xF0004811								0xF0004810					
															MODE														SEL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI CS Chip-Select and Mode

Field	Name	Description	
[0]	SEL	value	description
		0b0..001	Chip 0 selected for SPI Xfer.
		0b1..000	Chip N selected for SPI Xfer.
[16]	MODE	value	description
		0b0	Normal operation (CS handled by Core).
		0b1	Manual operation (CS handled by User, direct recopy of sel), useful for Bulk t

SPI_MASTER_LOOPBACK																													
0xF0004817								0xF0004816								0xF0004815								0xF0004814					
																													SEL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI CS Chip-Select and Mode

Field	Name	Description	
[0]	SEL	value	description
		0b0	Normal operation.
		0b1	Loopback operation (MOSI to MISO).

SPI_MASTER_CLK_DIVIDER																													
0xF000481B								0xF000481A								0xF0004819								0xF0004818					
SPI Clk Divider.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI Clk Divider.

The Timer is implemented as a countdown timer that can be used in various modes:
Polling : Returns current countdown value to software
One-Shot: Loads itself and stops when value reaches 0
Periodic: (Re-)Loads itself when value reaches 0

3.7. Timer

TIMER0_LOAD																											
0xF0005003								0xF0005002								0xF0005001								0xF0005000			
Timer Load value																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Load value when Timer is (re-)enabled. In One-Shot mode, the value written to this register specifies the Timer's duration in clock cycles.

TIMER0_RELOAD																											
0xF0005007								0xF0005006								0xF0005005								0xF0005004			
Timer Reload value																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reload value when Timer reaches 0. In Periodic mode, the value written to this register specify the Timer's period in clock cycles.

TIMER0_EN																											
0xF000500B								0xF000500A								0xF0005009								0xF0005008			
Timer Enable flag																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Enable flag of the Timer. Set this flag to 1 to enable/start the Timer. Set to 0 to disable the Timer.

TIMER0_UPDATE_VALUE																											
0xF000500F								0xF000500E								0xF000500D								0xF000500C			
Timer Update Value																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Update trigger for the current countdown value. A write to this register latches the current countdown value to value register.

TIMER0_VALUE																											
0xF0005013								0xF0005012								0xF0005011								0xF0005010			
Timer Value																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Latched countdown value. This value is updated by writing to update_value.

TIMER0_EV_STATUS																											
0xF0005017								0xF0005016								0xF0005015								0xF0005014			
																											ZERO
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the zero event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	ZERO	Level of the zero event

TIMER0_EV_PENDING																															
0xF000501B								0xF000501A								0xF0005019								0xF0005018							
																															ZERO
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a zero event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	ZERO	1 if a zero event occurred. This Event is triggered on a falling edge.

TIMER0_EV_ENABLE																															
0xF000501F								0xF000501E								0xF000501D								0xF000501C							
																															ZERO
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding zero events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	ZERO	Write a 1 to enable the zero Event

3.8. UART

UART_RXTX																															
0xF0005803								0xF0005802								0xF0005801								0xF0005800							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

It has the data for uart tx and rx, when reading from it it returns the value from RX and when writing to it writes to the TX

UART_TXFULL																															
0xF0005807								0xF0005806								0xF0005805								0xF0005804							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TX FIFO Full.

UART_RXEMPTY																															
0xF000580B								0xF000580A								0xF0005809								0xF0005808							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RX FIFO Empty.

UART_EV_STATUS																													
0xF000580F								0xF000580E								0xF000580D								0xF000580C					
(undefined, reads zero)																												RX	TX
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the rx event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	TX	Level of the tx event
[1]	RX	Level of the rx event

UART_EV_PENDING																													
0xF0005813								0xF0005812								0xF0005811								0xF0005810					
(undefined, reads zero)																												RX	TX
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a rx event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	TX	1 if a tx event occurred. This Event is triggered on a falling edge.
[1]	RX	1 if a rx event occurred. This Event is triggered on a falling edge.

UART_EV_ENABLE																													
0xF0005817								0xF0005816								0xF0005815								0xF0005814					
(undefined, reads zero)																												RX	TX
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding rx events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	TX	Write a 1 to enable the tx Event
[1]	RX	Write a 1 to enable the rx Event

UART_TXEMPTY																													
0xF000581B								0xF000581A								0xF0005819								0xF0005818					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TX FIFO Empty.

UART_RXFULL																													
0xF000581F								0xF000581E								0xF000581D								0xF000581C					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RX FIFO Full.

UART_ENABLED_OUT																															
0xF0006003								0xF0006002								0xF0006001								0xF0006000							
GPIO Output(s) Control.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Output(s) Control. *

3.9. IRQ

USER_IRQ_0_IN																															
0xF0006803								0xF0006802								0xF0006801								0xF0006800							
GPIO Input(s) Status.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_0_MODE																															
0xF0006807								0xF0006806								0xF0006805								0xF0006804							
GPIO IRQ Mode																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_0_EDGE																															
0xF000680B								0xF000680A								0xF0006809								0xF0006808							
GPIO IRQ Edge																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_0_EV_STATUS																															
0xF000680F								0xF000680E								0xF000680D								0xF000680C							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_0_EV_PENDING																															
0xF0006813								0xF0006812								0xF0006811								0xF0006810							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a I0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

USER_IRQ_0_EV_ENABLE																													
0xF0006817								0xF0006816								0xF0006815								0xF0006814					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_1_IN																													
0xF0007003								0xF0007002								0xF0007001								0xF0007000					
GPIO Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_1_MODE																													
0xF0007007								0xF0007006								0xF0007005								0xF0007004					
GPIO IRQ Mode																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_1_EDGE																													
0xF000700B								0xF000700A								0xF0007009								0xF0007008					
GPIO IRQ Edge																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_1_EV_STATUS																													
0xF000700F								0xF000700E								0xF000700D								0xF000700C					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_1_EV_PENDING																															
0xF0007013								0xF0007012								0xF0007011								0xF0007010							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a i0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	I0	1 if a i0 event occurred. This Event is triggered on a falling edge.

USER_IRQ_1_EV_ENABLE																															
0xF0007017								0xF0007016								0xF0007015								0xF0007014							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_2_IN																															
0xF0007803								0xF0007802								0xF0007801								0xF0007800							
GPIO Input(s) Status.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_2_MODE																															
0xF0007807								0xF0007806								0xF0007805								0xF0007804							
GPIO IRQ Mode																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_2_EDGE																															
0xF000780B								0xF000780A								0xF0007809								0xF0007808							
GPIO IRQ Edge																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_2_EV_STATUS																															
0xF000780F								0xF000780E								0xF000780D								0xF000780C							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_2_EV_PENDING																															
0xF0007813								0xF0007812								0xF0007811								0xF0007810							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a i0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	I0	1 if a i0 event occurred. This Event is triggered on a falling edge.

USER_IRQ_2_EV_ENABLE																															
0xF0007817								0xF0007816								0xF0007815								0xF0007814							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_3_IN																															
0xF0008003								0xF0008002								0xF0008001								0xF0008000							
GPIO Input(s) Status.																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_3_MODE																															
0xF0008007								0xF0008006								0xF0008005								0xF0008004							
GPIO IRQ Mode																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_3_EDGE																													
0xF000800B								0xF000800A								0xF0008009								0xF0008008					
GPIO IRQ Edge																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_3_EV_STATUS																													
0xF000800F								0xF000800E								0xF000800D								0xF000800C					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_3_EV_PENDING																													
0xF0008013								0xF0008012								0xF0008011								0xF0008010					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a i0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	I0	1 if a i0 event occurred. This Event is triggered on a falling edge.

USER_IRQ_3_EV_ENABLE																													
0xF0008017								0xF0008016								0xF0008015								0xF0008014					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_4_IN																													
0xF0008803								0xF0008802								0xF0008801								0xF0008800					
GPIO Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_4_MODE																													
0xF0008807								0xF0008806								0xF0008805								0xF0008804					
GPIO IRQ Mode																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_4_EDGE																													
0xF000880B								0xF000880A								0xF0008809								0xF0008808					
GPIO IRQ Edge																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_4_EV_STATUS																													
0xF000880F								0xF000880E								0xF000880D								0xF000880C					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_4_EV_PENDING																													
0xF0008813								0xF0008812								0xF0008811								0xF0008810					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a i0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	I0	1 if a i0 event occurred. This Event is triggered on a falling edge.

USER_IRQ_4_EV_ENABLE																													
0xF0008817								0xF0008816								0xF0008815								0xF0008814					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_5_IN																													
0xF0009003								0xF0009002								0xF0009001								0xF0009000					
GPIO Input(s) Status.																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO Input(s) Status. *

USER_IRQ_5_MODE																													
0xF0009007								0xF0009006								0xF0009005								0xF0009004					
GPIO IRQ Mode																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Mode: 0: Edge, 1: Change.

USER_IRQ_5_EDGE																													
0xF000900B								0xF000900A								0xF0009009								0xF0009008					
GPIO IRQ Edge																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPIO IRQ Edge (when in Edge mode): 0: Rising Edge, 1: Falling Edge.

USER_IRQ_5_EV_STATUS																													
0xF000900F								0xF000900E								0xF000900D								0xF000900C					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register contains the current raw level of the i0 event trigger. Writes to this register have no effect.

Field	Name	Description
[0]	I0	Level of the i0 event

USER_IRQ_5_EV_PENDING																													
0xF0009013								0xF0009012								0xF0009011								0xF0009010					
(undefined, reads zero)																													I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a i0 event occurs, the corresponding bit will be set in this register. To clear the Event, set the corresponding bit in this register.

Field	Name	Description
[0]	I0	1 if a i0 event occurred. This Event is triggered on a falling edge.

USER_IRQ_5_EV_ENABLE																															
0xF0009017								0xF0009016								0xF0009015								0xF0009014							
(undefined, reads zero)																															I0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding i0 events. Write a 0 to this register to disable individual events.

Field	Name	Description
[0]	I0	Write a 1 to enable the i0 Event

USER_IRQ_ENA_OUT																															
0xF0009803								0xF0009802								0xF0009801								0xF0009800							
(undefined, reads zero)																													ENA2	ENA1	ENA0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register enables the corresponding ENA signals. Write a 0 to this register to disable individual signals.

Field	Name	Description
[0]	ENA0	Write a 1 to enable the ENA0 Output
[1]	ENA1	Write a 1 to enable the ENA1 Output
[2]	ENA2	Write a 1 to enable the ENA2 Output

4. Address Space Mapping

The address space mapping is shown below.

Starting Address	Ending Address	Length	Acronym	Register Region / Memory Name
0x00000000	0x000003ff	0x00000400	dff	D Flip Flop Memory
0x00000400	0x000005ff	0x00000200	dff2	D Flip Flop Memory 2
0x10000000	0x10ffffff	0x01000000	flash	External Flash Memory Interface
0x30000000	0x3fffffff	0x10000000	mprj	User Project Register Space
0x26000000	0x262fffff	0x00300000	hk	HouseKeeping Register Space
0xf0000000	0xf000ffff	0x00010000	csr	SoC Control Register Space