PSoC® Creator™
Project Datasheet for Z80_3Chip

Creation Time: 09/06/2022 14:37:47
User: DESKTOP-2UF8LNB\HPz420
Project: Z80_3Chip
Tool: PSoC Creator 4.4

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
http://www.cypress.com

Copyright

Trademarks

Philips I2C Patent Rights

Purchase of I2C components from Cypress or one of its sublicensed Associated Companies conveys a license under the Philips I2C Patent Rights to use these components in an I2C system, provided that the system conforms to the I2C Standard Specification as defined by Philips. As from October 1st, 2006 Philips Semiconductors has a new trade name, NXP Semiconductors.

Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of a Cypress product in a life support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Datasheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as 'unbreakable.'

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Table of Contents

1 Overview

The Cypress PSoC 5 is a family of 32-bit devices with the following characteristics:

- High-performance 32-bit ARM Cortex-M3 core with a nested vectored interrupt controller (NVIC) and a high-performance DMA controller
- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals, such as USB, I2C and SPI
- Analog subsystem that includes 20-bit Delta Sigma converters (ADC), SAR ADCs, 8-bit DACs that can be configured for 12-bit operation, comparators, op amps and configurable switched capacitor (SC) and continuous time (CT) blocks to create PGAs, TIAs, mixers, and more
- Several types of memory elements, including SRAM, flash, and EEPROM
- Programming and debug system through JTAG, serial wire debug (SWD), and single wire viewer (SWV)
- Flexible routing to all pins

Figure 1 shows the major components of a typical CY8C52LP series member PSoC 5LP device. For details on all the systems listed above, please refer to the PSoC 5LP Technical Reference Manual .

Figure 1. CY8C52LP Device Series Block Diagram

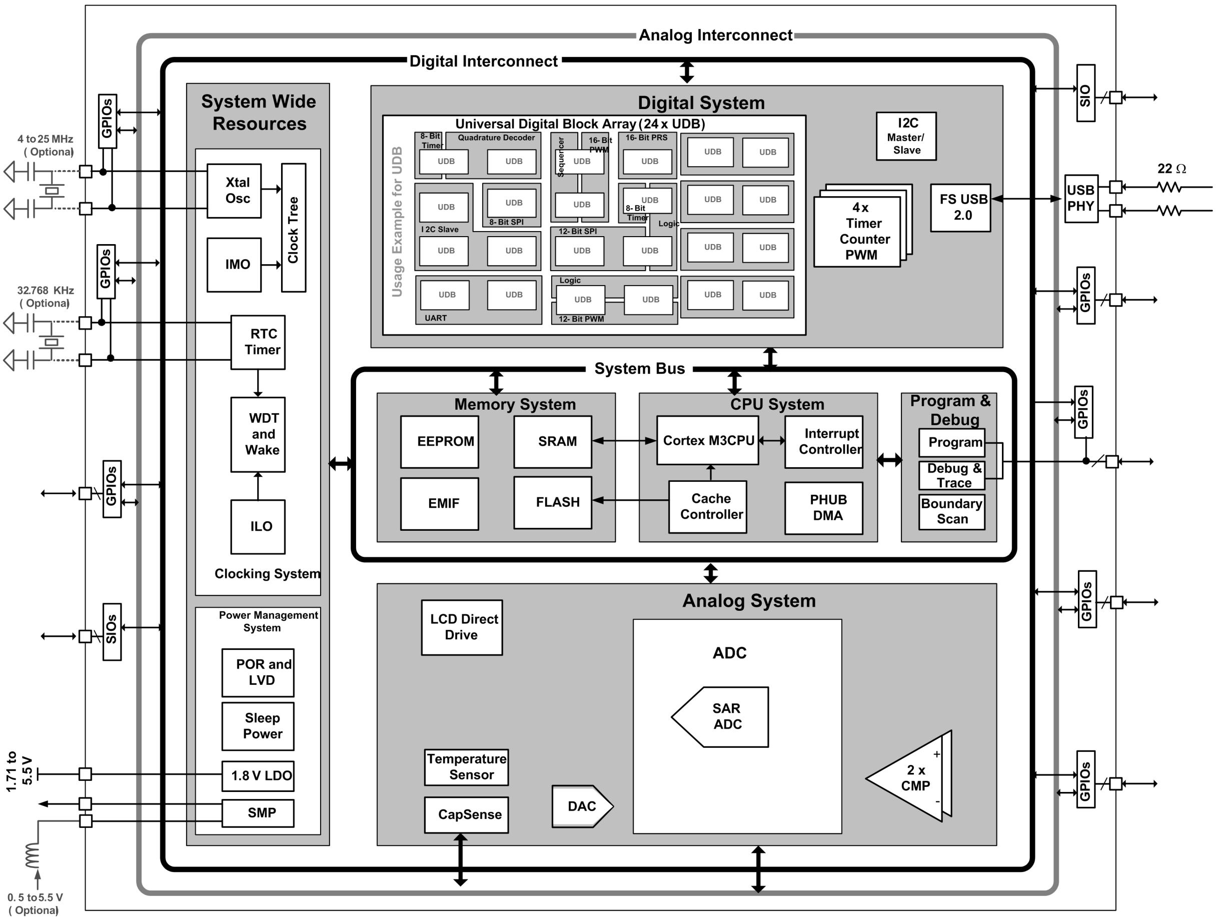


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C5267AXI-LP051
Package Name	100-TQFP
Family	PSoC 5LP
Series	CY8C52LP
Max CPU speed (MHz)	0
Flash size (kB)	128
SRAM size (kB)	32
EEPROM size (bytes)	2048
Vdd range (V)	1.71 to 5.5
Automotive qualified	No (Industrial Grade Only)
Temp range (Celsius)	-40 to 85
JTAG ID	0x2E133069

NOTE: The CPU speed noted above is the maximum available speed. The CPU is clocked by Bus Clock, listed in the System Clocks section below.

Table 2 lists the device resources that this design uses:

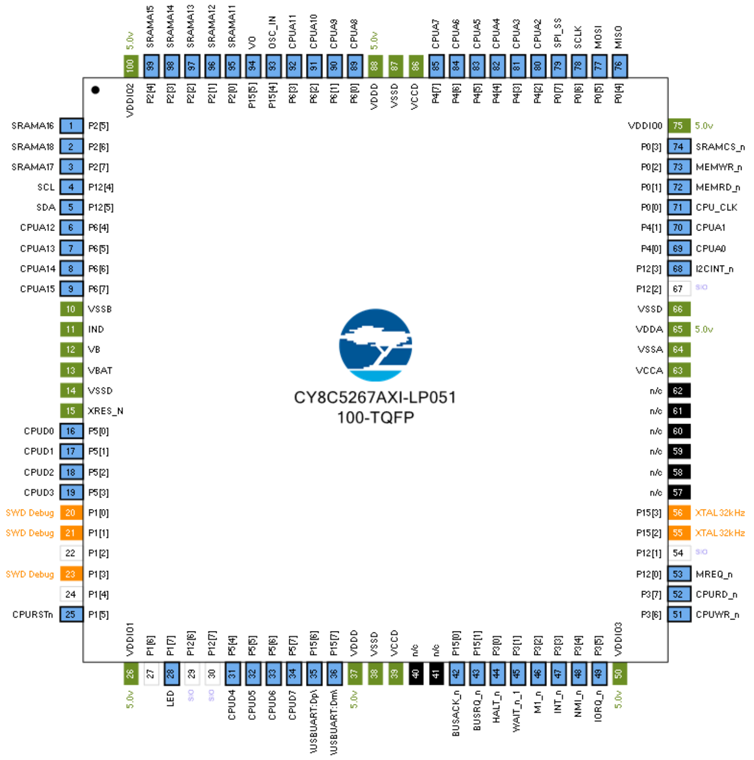
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Digital Clocks	4	4	8	50.00 %
Analog Clocks	0	4	4	0.00 %
CapSense Buffers	0	2	2	0.00 %
Interrupts	12	20	32	37.50 %
IO	65	7	72	90.28 %
Segment LCD	0	1	1	0.00 %
I2C	1	0	1	100.00 %
USB	1	0	1	100.00 %
DMA Channels	2	22	24	8.33 %
Timer	1	3	4	25.00 %
UDB				
Macrocells	55	137	192	28.65 %
Unique P-terms	138	246	384	35.94 %
Total P-terms	144			
Datapath Cells	4	20	24	16.67 %
Status Cells	8	16	24	33.33 %
Status Registers	3			
StatusI Registers	3			
Sync Cells (x2)	1			
Routed Count7 Load/Enable	1			
Control Cells	17	7	24	70.83 %
Control Registers	16			
Count7 Cells	1			
Comparator	0	2	2	0.00 %
Delta-Sigma ADC	0	1	1	0.00 %
LPF	0	2	2	0.00 %
SAR ADC	0	1	1	0.00 %
DAC				
VIDAC	1	0	1	100.00 %

2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



2.1 Hardware Pins

Table 3 contains information about the pins on this device in device pin order. (No connection ["n/c"] pins have been omitted.)

Table 3. Device Pins

Pin	Port	Name	Type	Drive Mode	Reset State
1	P2[5]	SRAMA16	Dgtl Out	Strong drive	HiZ Analog Unb
2	P2[6]	SRAMA18	Dgtl Out	Strong drive	HiZ Analog Unb
3	P2[7]	SRAMA17	Dgtl Out	Strong drive	HiZ Analog Unb
4	P12[4]	SCL	Dgtl I/O	OD, DL	Pulled Up
5	P12[5]	SDA	Dgtl I/O	OD, DL	Pulled Up
6	P6[4]	CPUA12	Dgtl In	HiZ digital	HiZ Analog Unb
7	P6[5]	CPUA13	Dgtl In	HiZ digital	HiZ Analog Unb
8	P6[6]	CPUA14	Dgtl In	HiZ digital	HiZ Analog Unb
9	P6[7]	CPUA15	Dgtl In	HiZ digital	HiZ Analog Unb
10	VSSB	VSSB	Dedicated		
11	IND	IND	Dedicated		
12	VB	VB	Dedicated		
13	VBAT	VBAT	Dedicated		
14	VSSD	VSSD	Power		
15	XRES_N	XRES_N	Dedicated		
16	P5[0]	CPUD0	Dgtl I/O	Res pull up	HiZ Analog Unb
17	P5[1]	CPUD1	Dgtl I/O	Res pull up	HiZ Analog Unb
18	P5[2]	CPUD2	Dgtl I/O	Res pull up	HiZ Analog Unb
19	P5[3]	CPUD3	Dgtl I/O	Res pull up	HiZ Analog Unb
20	P1[0]	Debug:SWD_IO	Reserved		
21	P1[1]	Debug:SWD_CK	Reserved		
22	P1[2]	GPIO [unused]			HiZ Analog Unb
23	P1[3]	Debug:SWV	Reserved		
24	P1[4]	GPIO [unused]			HiZ Analog Unb
25	P1[5]	CPURSTn	Dgtl Out	Strong drive	HiZ Analog Unb
26	VDDIO1	VDDIO1	Power		
27	P1[6]	GPIO [unused]			HiZ Analog Unb
28	P1[7]	LED	Software In/Out	Strong drive	HiZ Analog Unb
29	P12[6]	SIO [unused]			Pulled Up
30	P12[7]	SIO [unused]			Pulled Up
31	P5[4]	CPUD4	Dgtl I/O	Res pull up	HiZ Analog Unb
32	P5[5]	CPUD5	Dgtl I/O	Res pull up	HiZ Analog Unb
33	P5[6]	CPUD6	Dgtl I/O	Res pull up	HiZ Analog Unb
34	P5[7]	CPUD7	Dgtl I/O	Res pull up	HiZ Analog Unb
35	P15[6]	\USBUART:Dp\	Analog	HiZ analog	HiZ Analog Unb
36	P15[7]	\USBUART:Dm\	Analog	HiZ analog	HiZ Analog Unb
37	VDDD	VDDD	Power		
38	VSSD	VSSD	Power		
39	VCCD	VCCD	Power		
42	P15[0]	BUSACK_n	Software In/Out	Res pull up	HiZ Analog Unb
43	P15[1]	BUSRQ_n	Software In/Out	Strong drive	HiZ Analog Unb
44	P3[0]	HALT_n	Software In/Out	Res pull up	HiZ Analog Unb

Pin	Port	Name	Type	Drive Mode	Reset State
45	P3[1]	WAIT_n_1	Dgtl Out	Strong drive	HiZ Analog Unb
46	P3[2]	M1_n	Dgtl In	Res pull up	HiZ Analog Unb
47	P3[3]	INT_n	Software In/Out	Strong drive	HiZ Analog Unb
48	P3[4]	NMI_n	Dgtl Out	Strong drive	HiZ Analog Unb
49	P3[5]	IORQ_n	Dgtl In	Res pull up	HiZ Analog Unb
50	VDDIO3	VDDIO3	Power		
51	P3[6]	CPUWR_n	Dgtl In	Res pull up	HiZ Analog Unb
52	P3[7]	CPURD_n	Dgtl In	Res pull up	HiZ Analog Unb
53	P12[0]	MREQ_n	Dgtl In	Res pull up	Pulled Up
54	P12[1]	SIO [unused]			Pulled Up
55	P15[2]	XTAL 32kHz:Xi	Reserved		
56	P15[3]	XTAL 32kHz:Xi	Reserved		
63	VCCA	VCCA	Power		
64	VSSA	VSSA	Power		
65	VDDA	VDDA	Power		
66	VSSD	VSSD	Power		
67	P12[2]	SIO [unused]			Pulled Up
68	P12[3]	I2CINT_n	Dgtl In	Res pull up	Pulled Up
69	P4[0]	CPUA0	Dgtl I/O	Strong drive	HiZ Analog Unb
70	P4[1]	CPUA1	Dgtl I/O	Strong drive	HiZ Analog Unb
71	P0[0]	CPU_CLK	Dgtl Out	Strong drive	HiZ Analog Unb
72	P0[1]	MEMRD_n	Dgtl Out	Strong drive	HiZ Analog Unb
73	P0[2]	MEMWR_n	Dgtl Out	Strong drive	HiZ Analog Unb
74	P0[3]	SRAMCS_n	Dgtl Out	Strong drive	HiZ Analog Unb
75	VDDIO0	VDDIO0	Power		
76	P0[4]	MISO	Dgtl In	Res pull up	HiZ Analog Unb
77	P0[5]	MOSI	Dgtl Out	Strong drive	HiZ Analog Unb
78	P0[6]	SCLK	Dgtl Out	Strong drive	HiZ Analog Unb
79	P0[7]	SPI_SS	Dgtl Out	Strong drive	HiZ Analog Unb
80	P4[2]	CPUA2	Dgtl I/O	Strong drive	HiZ Analog Unb
81	P4[3]	CPUA3	Dgtl I/O	Strong drive	HiZ Analog Unb
82	P4[4]	CPUA4	Dgtl I/O	Strong drive	HiZ Analog Unb
83	P4[5]	CPUA5	Dgtl I/O	Strong drive	HiZ Analog Unb
84	P4[6]	CPUA6	Dgtl I/O	Strong drive	HiZ Analog Unb
85	P4[7]	CPUA7	Dgtl I/O	Strong drive	HiZ Analog Unb
86	VCCD	VCCD	Power		
87	VSSD	VSSD	Power		
88	VDDD	VDDD	Power		
89	P6[0]	CPUA8	Dgtl Out	Res pull up	HiZ Analog Unb
90	P6[1]	CPUA9	Dgtl Out	Res pull up	HiZ Analog Unb
91	P6[2]	CPUA10	Dgtl Out	Res pull up	HiZ Analog Unb
92	P6[3]	CPUA11	Dgtl In	HiZ digital	HiZ Analog Unb
93	P15[4]	OSC_IN	Software In/Out	HiZ digital	HiZ Analog Unb
94	P15[5]	VO	Analog	HiZ analog	HiZ Analog Unb
95	P2[0]	SRAMA11	Dgtl Out	Strong drive	HiZ Analog Unb
96	P2[1]	SRAMA12	Dgtl Out	Strong drive	HiZ Analog Unb
97	P2[2]	SRAMA13	Dgtl Out	Strong drive	HiZ Analog Unb
98	P2[3]	SRAMA14	Dgtl Out	Strong drive	HiZ Analog Unb
99	P2[4]	SRAMA15	Dgtl Out	Strong drive	HiZ Analog Unb
100	VDDIO2	VDDIO2	Power		

Abbreviations used in Table 3 have the following meanings:

- Dgtl Out = Digital Output
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl I/O = Digital In/Out
- OD, DL = Open drain, drives low
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Res pull up = Resistive pull up
- HiZ analog = High impedance analog

2.2 Hardware Ports

Table 4 contains information about the pins on this device in device port order. (No connection ["n/c"], power and dedicated pins have been omitted.)

Table 4. Device Ports

Port	Pin	Name	Type	Drive Mode	Reset State
P0[0]	71	CPU_CLK	Dgtl Out	Strong drive	HiZ Analog Unb
P0[1]	72	MEMRD_n	Dgtl Out	Strong drive	HiZ Analog Unb
P0[2]	73	MEMWR_n	Dgtl Out	Strong drive	HiZ Analog Unb
P0[3]	74	SRAMCS_n	Dgtl Out	Strong drive	HiZ Analog Unb
P0[4]	76	MISO	Dgtl In	Res pull up	HiZ Analog Unb
P0[5]	77	MOSI	Dgtl Out	Strong drive	HiZ Analog Unb
P0[6]	78	SCLK	Dgtl Out	Strong drive	HiZ Analog Unb
P0[7]	79	SPI_SS	Dgtl Out	Strong drive	HiZ Analog Unb
P1[0]	20	Debug:SWD_IO	Reserved		
P1[1]	21	Debug:SWD_CK	Reserved		
P1[2]	22	GPIO [unused]			HiZ Analog Unb
P1[3]	23	Debug:SWV	Reserved		
P1[4]	24	GPIO [unused]			HiZ Analog Unb
P1[5]	25	CPURSTn	Dgtl Out	Strong drive	HiZ Analog Unb
P1[6]	27	GPIO [unused]			HiZ Analog Unb
P1[7]	28	LED	Software In/Out	Strong drive	HiZ Analog Unb
P12[0]	53	MREQ_n	Dgtl In	Res pull up	Pulled Up
P12[1]	54	SIO [unused]			Pulled Up
P12[2]	67	SIO [unused]			Pulled Up
P12[3]	68	I2CINT_n	Dgtl In	Res pull up	Pulled Up
P12[4]	4	SCL	Dgtl I/O	OD, DL	Pulled Up
P12[5]	5	SDA	Dgtl I/O	OD, DL	Pulled Up
P12[6]	29	SIO [unused]			Pulled Up
P12[7]	30	SIO [unused]			Pulled Up
P15[0]	42	BUSACK_n	Software In/Out	Res pull up	HiZ Analog Unb
P15[1]	43	BUSRQ_n	Software In/Out	Strong drive	HiZ Analog Unb
P15[2]	55	XTAL 32kHz:Xi	Reserved		
P15[3]	56	XTAL 32kHz:Xi	Reserved		
P15[4]	93	OSC_IN	Software In/Out	HiZ digital	HiZ Analog Unb
P15[5]	94	VO	Analog	HiZ analog	HiZ Analog Unb
P15[6]	35	\USBUART:Dp\	Analog	HiZ analog	HiZ Analog Unb
P15[7]	36	\USBUART:Dm\	Analog	HiZ analog	HiZ Analog Unb
P2[0]	95	SRAMA11	Dgtl Out	Strong drive	HiZ Analog Unb
P2[1]	96	SRAMA12	Dgtl Out	Strong drive	HiZ Analog Unb
P2[2]	97	SRAMA13	Dgtl Out	Strong drive	HiZ Analog Unb
P2[3]	98	SRAMA14	Dgtl Out	Strong drive	HiZ Analog Unb
P2[4]	99	SRAMA15	Dgtl Out	Strong drive	HiZ Analog Unb
P2[5]	1	SRAMA16	Dgtl Out	Strong drive	HiZ Analog Unb
P2[6]	2	SRAMA18	Dgtl Out	Strong drive	HiZ Analog Unb
P2[7]	3	SRAMA17	Dgtl Out	Strong drive	HiZ Analog Unb
P3[0]	44	HALT_n	Software In/Out	Res pull up	HiZ Analog Unb

Port	Pin	Name	Type	Drive Mode	Reset State
P3[1]	45	WAIT_n_1	Dgtl Out	Strong drive	HiZ Analog Unb
P3[2]	46	M1_n	Dgtl In	Res pull up	HiZ Analog Unb
P3[3]	47	INT_n	Software In/Out	Strong drive	HiZ Analog Unb
P3[4]	48	NMI_n	Dgtl Out	Strong drive	HiZ Analog Unb
P3[5]	49	IORQ_n	Dgtl In	Res pull up	HiZ Analog Unb
P3[6]	51	CPUWR_n	Dgtl In	Res pull up	HiZ Analog Unb
P3[7]	52	CPURD_n	Dgtl In	Res pull up	HiZ Analog Unb
P4[0]	69	CPUA0	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[1]	70	CPUA1	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[2]	80	CPUA2	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[3]	81	CPUA3	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[4]	82	CPUA4	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[5]	83	CPUA5	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[6]	84	CPUA6	Dgtl I/O	Strong drive	HiZ Analog Unb
P4[7]	85	CPUA7	Dgtl I/O	Strong drive	HiZ Analog Unb
P5[0]	16	CPUD0	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[1]	17	CPUD1	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[2]	18	CPUD2	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[3]	19	CPUD3	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[4]	31	CPUD4	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[5]	32	CPUD5	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[6]	33	CPUD6	Dgtl I/O	Res pull up	HiZ Analog Unb
P5[7]	34	CPUD7	Dgtl I/O	Res pull up	HiZ Analog Unb
P6[0]	89	CPUA8	Dgtl Out	Res pull up	HiZ Analog Unb
P6[1]	90	CPUA9	Dgtl Out	Res pull up	HiZ Analog Unb
P6[2]	91	CPUA10	Dgtl Out	Res pull up	HiZ Analog Unb
P6[3]	92	CPUA11	Dgtl In	HiZ digital	HiZ Analog Unb
P6[4]	6	CPUA12	Dgtl In	HiZ digital	HiZ Analog Unb
P6[5]	7	CPUA13	Dgtl In	HiZ digital	HiZ Analog Unb
P6[6]	8	CPUA14	Dgtl In	HiZ digital	HiZ Analog Unb
P6[7]	9	CPUA15	Dgtl In	HiZ digital	HiZ Analog Unb

Abbreviations used in Table 4 have the following meanings:

- Dgtl Out = Digital Output
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl In = Digital Input
- Res pull up = Resistive pull up
- Dgtl I/O = Digital In/Out
- OD, DL = Open drain, drives low
- HiZ digital = High impedance digital
- HiZ analog = High impedance analog

2.3 Software Pins

Table 5 contains information about the software pins on this device in alphabetical order. (Only software-accessible pins are shown.)

Table 5. Software Pins

Name	Port	Type	Reset State
\USBUART:Dm\	P15[7]	Analog	HiZ Analog Unb
\USBUART:Dp\	P15[6]	Analog	HiZ Analog Unb
BUSACK_n	P15[0]	Software In/Out	HiZ Analog Unb
BUSRQ_n	P15[1]	Software In/Out	HiZ Analog Unb
CPU_CLK	P0[0]	Dgtl Out	HiZ Analog Unb
CPUA0	P4[0]	Dgtl I/O	HiZ Analog Unb
CPUA1	P4[1]	Dgtl I/O	HiZ Analog Unb
CPUA10	P6[2]	Dgtl Out	HiZ Analog Unb
CPUA11	P6[3]	Dgtl In	HiZ Analog Unb
CPUA12	P6[4]	Dgtl In	HiZ Analog Unb
CPUA13	P6[5]	Dgtl In	HiZ Analog Unb
CPUA14	P6[6]	Dgtl In	HiZ Analog Unb
CPUA15	P6[7]	Dgtl In	HiZ Analog Unb
CPUA2	P4[2]	Dgtl I/O	HiZ Analog Unb
CPUA3	P4[3]	Dgtl I/O	HiZ Analog Unb
CPUA4	P4[4]	Dgtl I/O	HiZ Analog Unb
CPUA5	P4[5]	Dgtl I/O	HiZ Analog Unb
CPUA6	P4[6]	Dgtl I/O	HiZ Analog Unb
CPUA7	P4[7]	Dgtl I/O	HiZ Analog Unb
CPUA8	P6[0]	Dgtl Out	HiZ Analog Unb
CPUA9	P6[1]	Dgtl Out	HiZ Analog Unb
CPUD0	P5[0]	Dgtl I/O	HiZ Analog Unb
CPUD1	P5[1]	Dgtl I/O	HiZ Analog Unb
CPUD2	P5[2]	Dgtl I/O	HiZ Analog Unb
CPUD3	P5[3]	Dgtl I/O	HiZ Analog Unb
CPUD4	P5[4]	Dgtl I/O	HiZ Analog Unb
CPUD5	P5[5]	Dgtl I/O	HiZ Analog Unb
CPUD6	P5[6]	Dgtl I/O	HiZ Analog Unb
CPUD7	P5[7]	Dgtl I/O	HiZ Analog Unb
CPURD_n	P3[7]	Dgtl In	HiZ Analog Unb
CPURSTn	P1[5]	Dgtl Out	HiZ Analog Unb
CPUWR_n	P3[6]	Dgtl In	HiZ Analog Unb
Debug:SWD_CK	P1[1]	Reserved	
Debug:SWD_IO	P1[0]	Reserved	
Debug:SWV	P1[3]	Reserved	
GPIO [unused]	P1[4]		HiZ Analog Unb
GPIO [unused]	P1[2]		HiZ Analog Unb
GPIO [unused]	P1[6]		HiZ Analog Unb
HALT_n	P3[0]	Software In/Out	HiZ Analog Unb
I2CINT_n	P12[3]	Dgtl In	Pulled Up
INT_n	P3[3]	Software In/Out	HiZ Analog Unb
IORQ_n	P3[5]	Dgtl In	HiZ Analog Unb

Name	Port	Type	Reset State
LED	P1[7]	Software In/Out	HiZ Analog Unb
M1_n	P3[2]	Dgtl In	HiZ Analog Unb
MEMRD_n	P0[1]	Dgtl Out	HiZ Analog Unb
MEMWR_n	P0[2]	Dgtl Out	HiZ Analog Unb
MISO	P0[4]	Dgtl In	HiZ Analog Unb
MOSI	P0[5]	Dgtl Out	HiZ Analog Unb
MREQ_n	P12[0]	Dgtl In	Pulled Up
NMI_n	P3[4]	Dgtl Out	HiZ Analog Unb
OSC_IN	P15[4]	Software In/Out	HiZ Analog Unb
SCL	P12[4]	Dgtl I/O	Pulled Up
SCLK	P0[6]	Dgtl Out	HiZ Analog Unb
SDA	P12[5]	Dgtl I/O	Pulled Up
SIO [unused]	P12[7]		Pulled Up
SIO [unused]	P12[6]		Pulled Up
SIO [unused]	P12[1]		Pulled Up
SIO [unused]	P12[2]		Pulled Up
SPI_SS	P0[7]	Dgtl Out	HiZ Analog Unb
SRAMA11	P2[0]	Dgtl Out	HiZ Analog Unb
SRAMA12	P2[1]	Dgtl Out	HiZ Analog Unb
SRAMA13	P2[2]	Dgtl Out	HiZ Analog Unb
SRAMA14	P2[3]	Dgtl Out	HiZ Analog Unb
SRAMA15	P2[4]	Dgtl Out	HiZ Analog Unb
SRAMA16	P2[5]	Dgtl Out	HiZ Analog Unb
SRAMA17	P2[7]	Dgtl Out	HiZ Analog Unb
SRAMA18	P2[6]	Dgtl Out	HiZ Analog Unb
SRAMCS_n	P0[3]	Dgtl Out	HiZ Analog Unb
VO	P15[5]	Analog	HiZ Analog Unb
WAIT_n_1	P3[1]	Dgtl Out	HiZ Analog Unb
XTAL 32kHz:Xi	P15[2]	Reserved	
XTAL 32kHz:Xi	P15[3]	Reserved	

Abbreviations used in Table 5 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl Out = Digital Output
- Dgtl I/O = Digital In/Out
- Dgtl In = Digital Input

For more information on reading, writing and configuring pins, please refer to:

- Pins chapter in the System Reference Guide
  - CyPins API routines
- Programming Application Interface section in the cy_pins component datasheet

3 System Settings

3.1 System Configuration

Table 6. System Configuration Settings

Name	Value
Device Configuration Mode	Compressed
Enable Error Correcting Code (ECC)	False
Store Configuration Data in ECC Memory	True
Instruction Cache Enabled	True
Enable Fast IMO During Startup	True
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x200
Stack Size (bytes)	0x0800
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 7. System Debug Settings

Name	Value
Debug Select	SWD+SWV (serial wire debug and viewer)
Enable Device Protection	False
Embedded Trace (ETM)	False
Use Optional XRES	False

3.3 System Operating Conditions

Table 8. System Operating Conditions

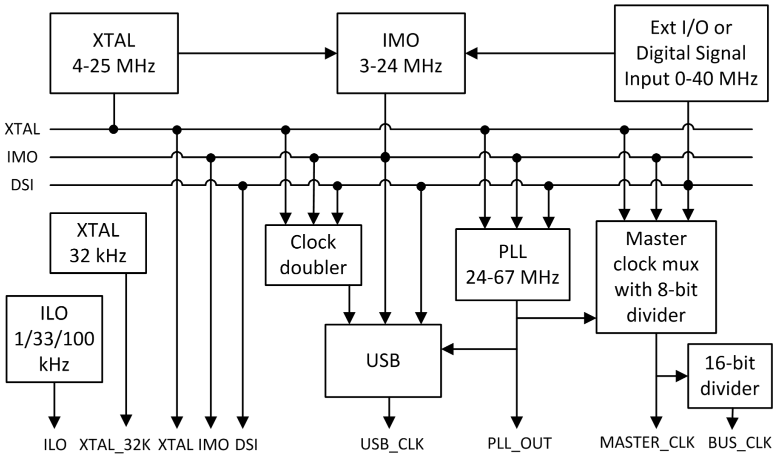
Name	Value
VDDA (V)	5.0
VDDD (V)	5.0
VDDIO0 (V)	5.0
VDDIO1 (V)	5.0
VDDIO2 (V)	5.0
VDDIO3 (V)	5.0
Variable VDDA	False
Temperature Range	-40C - 85/125C

4 Clocks

The clock system includes these clock resources:

- Four internal clock sources increase system integration:
  - o 3 to 74.7 MHz Internal Main Oscillator (IMO) ±1% at 3 MHz
  - o 1 kHz, 33 kHz, and 100 kHz Internal Low Speed Oscillator (ILO) outputs
  - o 12 to 80 MHz clock doubler output, sourced from IMO, MHz External Crystal Oscillator (MHzECO), and Digital System Interconnect (DSI)
  - o 24 to 80 MHz fractional Phase-Locked Loop (PLL) sourced from IMO, MHzECO, and DSI
- Clock generated using a DSI signal from an external I/O pin or other logic
- Two external clock sources provide high precision clocks:
  - o 4 to 25 MHz External Crystal Oscillator (MHzECO)
  - o 32.768 kHz External Crystal Oscillator (kHzECO) for Real Time Clock (RTC)
- Dedicated 16-bit divider for bus clock
- Eight individually sourced 16-bit clock dividers for the digital system peripherals
- Four individually sourced 16-bit clock dividers with skew for the analog system peripherals
- IMO has a USB mode that synchronizes to USB host traffic, requiring no external crystal for USB. (USB equipped parts only)

Figure 3. System Clock Configuration



4.1 System Clocks

Table 9 lists the system clocks used in this design.

Table 9. System Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
BUS_CLK	DIGITAL	MASTER_CLK	? MHz	60 MHz	±0.25	True	True
MASTER_CLK	DIGITAL	PLL_OUT	? MHz	60 MHz	±0.25	True	True
PLL_OUT	DIGITAL	IMO	60 MHz	60 MHz	±0.25	True	True
USB_CLK	DIGITAL	IMO	48 MHz	48 MHz	±0.25	False	True
IMO	DIGITAL		24 MHz	24 MHz	±0.25	True	True
ILO	DIGITAL		? MHz	100 kHz	-55,+100	True	True
XTAL 32kHz	DIGITAL		32.768 kHz	32.768 kHz	±0	False	True
Digital Signal	DIGITAL		? MHz	? MHz	±0	False	False
XTAL	DIGITAL		24 MHz	? MHz	±0	False	False

4.2 Local and Design Wide Clocks

Local clocks drive individual analog and digital blocks. Design wide clocks are a user-defined optimization, where two or more analog or digital blocks that share a common clock profile (frequency, etc) can be driven from the same clock divider output source.

Figure 4. Local and Design Wide Clock Configuration

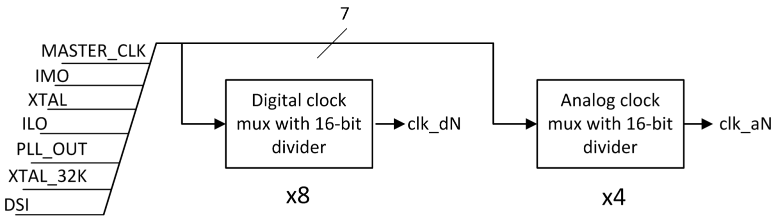

7
MASTER_CLK
IMO
XTAL
ILO
PLL_OUT
XTAL_32K
DSI
Digital clock mux with 16-bit divider
clk_dN
x8
Analog clock mux with 16-bit divider
clk_aN
x4

Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
timer_clock	DIGITAL	BUS_CLK	? MHz	60 MHz	±0.25	True	True
I2C_BusClock	DIGITAL	BUS_CLK	? MHz	60 MHz	±0.25	True	True
Clock_1	DIGITAL	MASTER_CLK	? MHz	60 MHz	±0.25	True	True
DAC_Clock	DIGITAL	MASTER_CLK	? MHz	30 MHz	±0.25	True	True
CPUCLK	DIGITAL	PLL_OUT	12.5 MHz	12 MHz	±0.25	True	True
SPI_Master_-IntClock	DIGITAL	MASTER_CLK	800 kHz	800 kHz	±0.25	True	True

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 5LP Technical Reference Manual
- Clocking chapter in the System Reference Guide
  - CyPLL API routines
  - CyIMO API routines

- CyILO API routines
- CyMaster API routines
- CyXTAL API routines

5 Interrupts and DMAs

5.1 Interrupts

This design contains the following interrupt components: (0 is the highest priority)

Table 11. Interrupts

Name	Intr Num	Vector	Priority
I2CINT_ISR	0	0	7
Timer_ISR	1	1	7
RTC_isr	2	2	7
USBUART_ep_1	3	3	7
USBUART_ep_2	4	4	7
USBUART_ep_3	5	5	7
USBUART_dp_int	12	12	7
I2C_I2C_IRQ	15	15	7
USBUART_sof_int	21	21	7
USBUART_arb_int	22	22	7
USBUART_bus_reset	23	23	7
USBUART_ep_0	24	24	7

For more information on interrupts, please refer to:

- Interrupt Controller chapter in the PSoC 5LP Technical Reference Manual
- Interrupts chapter in the System Reference Guide
  - CyInt API routines and related registers
- Datasheet for cy_isr component

5.2 DMAs

This design contains the following DMA components: (0 is the highest priority)

Table 12. DMAs

Name	Priority	Channel Number
WaveDAC8_Wave1_DMA	2	0
WaveDAC8_Wave2_DMA	2	1

For more information on DMAs, please refer to:

- PHUB and DMAC chapter in the PSoC 5LP Technical Reference Manual
- DMA chapter in the System Reference Guide
  - DMA API routines and related registers
- Datasheet for cy_dma component

6 Flash Memory

PSoC 5LP devices offer a host of Flash protection options and device security features that you can leverage to meet the security and protection requirements of an application. These requirements range from protecting configuration settings or Flash data to locking the entire device from external access.

Table 13 lists the Flash protection settings for your design.

Table 13. Flash Protection Settings

Start Address	End Address	Protection Level
0x0	0x1FFFF	U - Unprotected

Flash memory is organized as rows with each row of flash having 256 bytes. Each flash row can be assigned one of four protection levels:

- U - Unprotected
- F - Factory Upgrade
- R - Field Upgrade
- W - Full Protection

For more information on Flash memory and protection, please refer to:

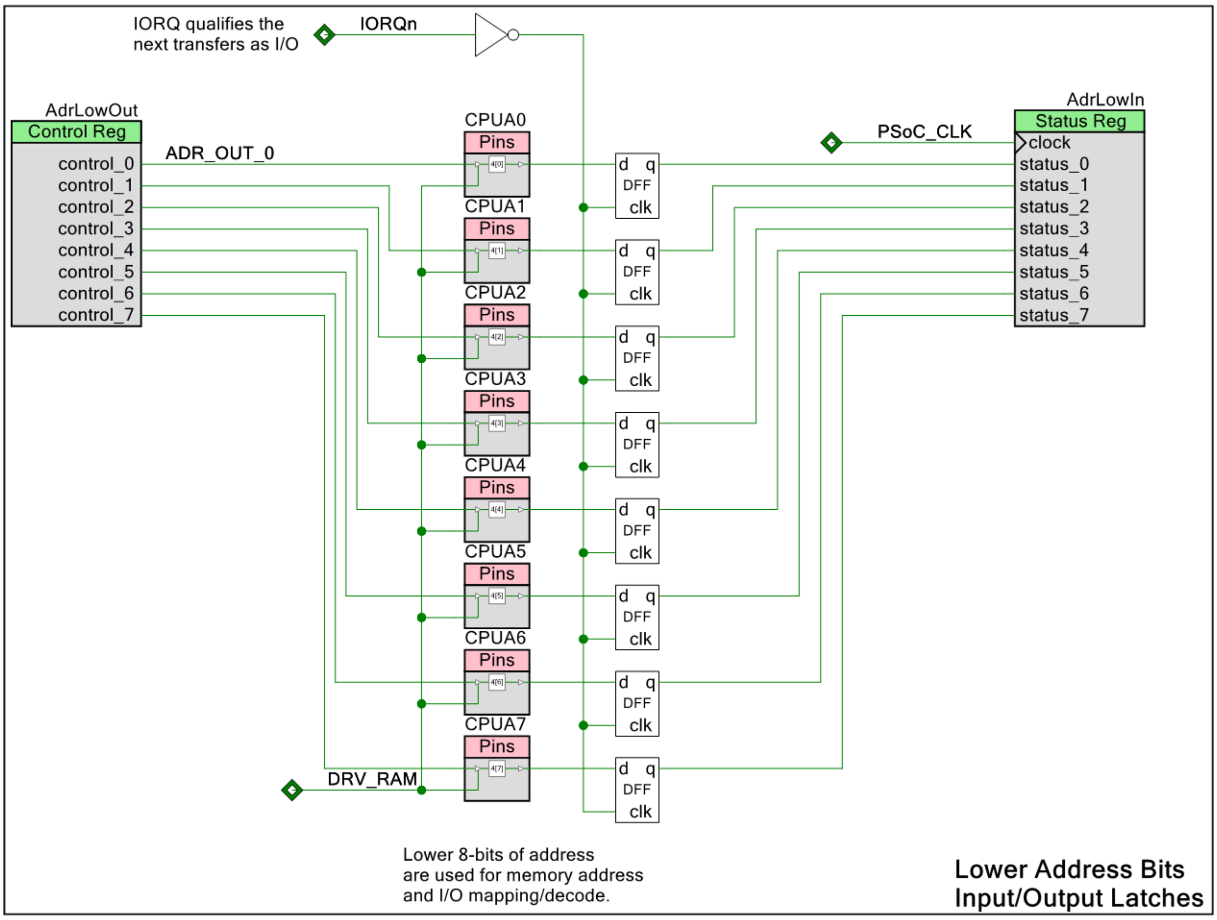
- Flash Protection chapter in the PSoC 5LP Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
    - CyWrite API routines
    - CyFlash API routines

7 Design Contents

This design's schematic content consists of the following 11 schematic sheets:

7.1 Schematic Sheet: AddrLower

Figure 5. Schematic Sheet: AddrLower

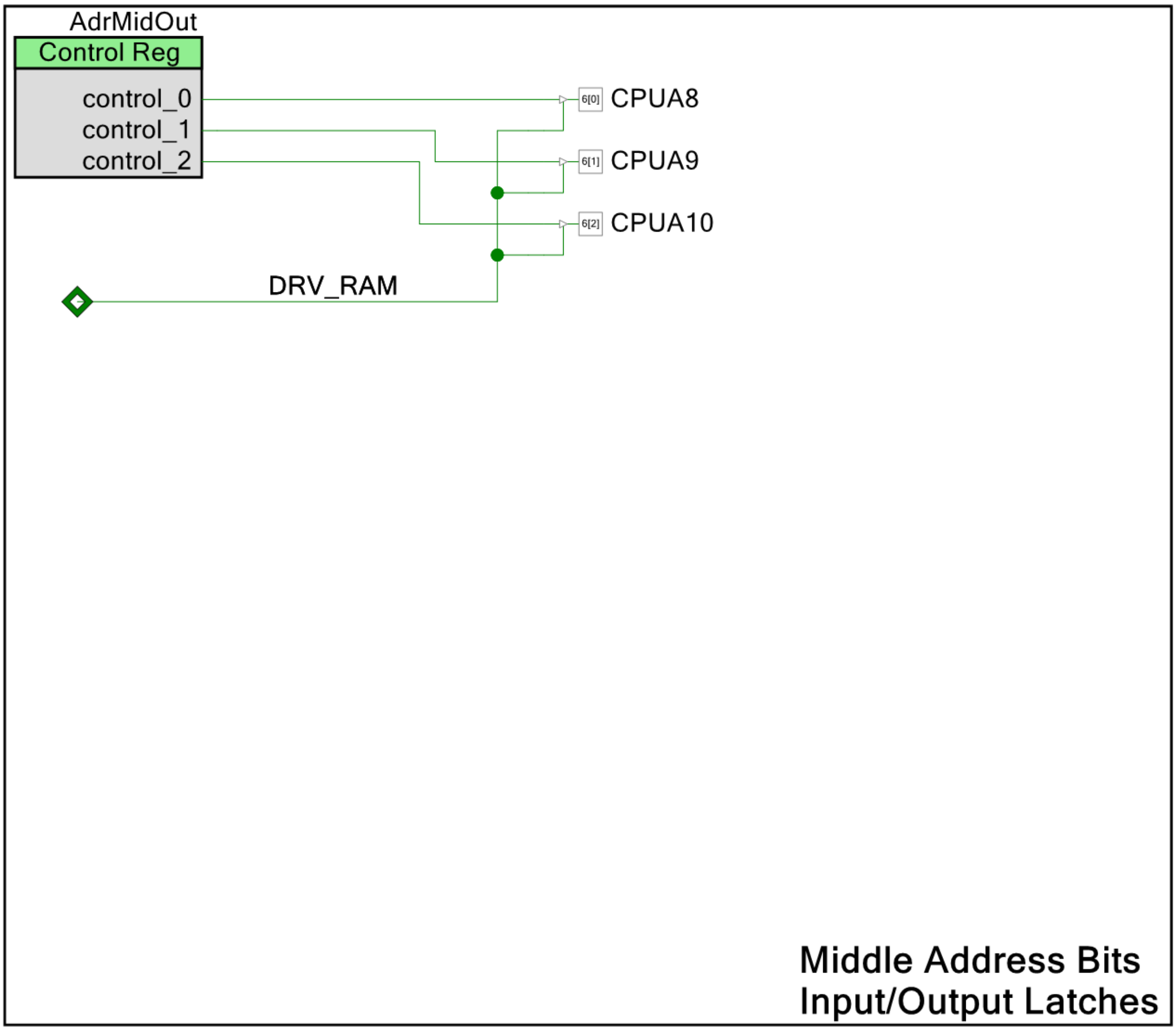


This schematic sheet contains the following component instances:

- Instance AdrLowIn (type: CyStatusReg_v1_90)
- Instance AdrLowOut (type: CyControlReg_v1_80)
- Instance cydff_1 (type: cydff_v1_30)
- Instance cydff_2 (type: cydff_v1_30)
- Instance cydff_3 (type: cydff_v1_30)
- Instance cydff_4 (type: cydff_v1_30)
- Instance cydff_5 (type: cydff_v1_30)
- Instance cydff_6 (type: cydff_v1_30)
- Instance cydff_7 (type: cydff_v1_30)
- Instance cydff_8 (type: cydff_v1_30)

7.2 Schematic Sheet: AddressMid

Figure 6. Schematic Sheet: AddressMid

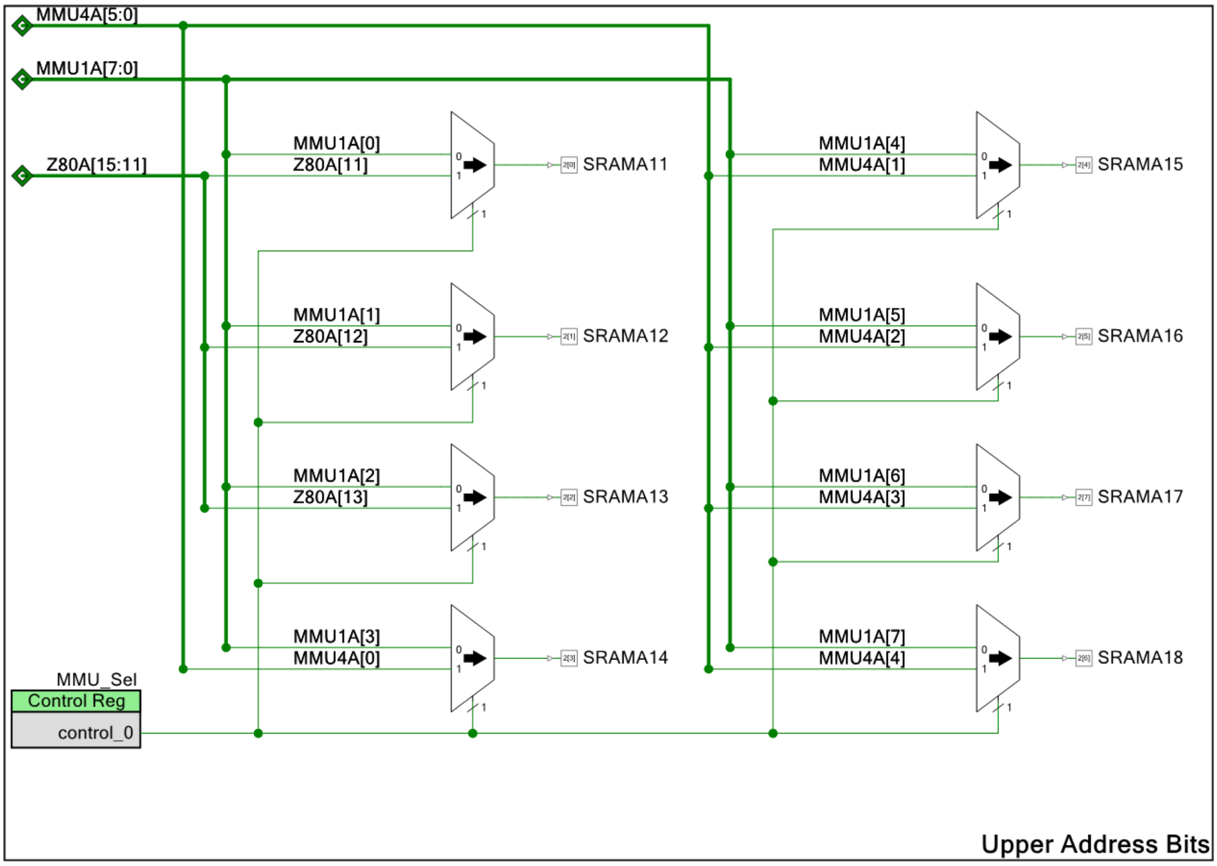


This schematic sheet contains the following component instances:

- Instance AdrMidOut (type: CyControlReg_v1_80)

7.3 Schematic Sheet: AddressUpper

Figure 7. Schematic Sheet: AddressUpper

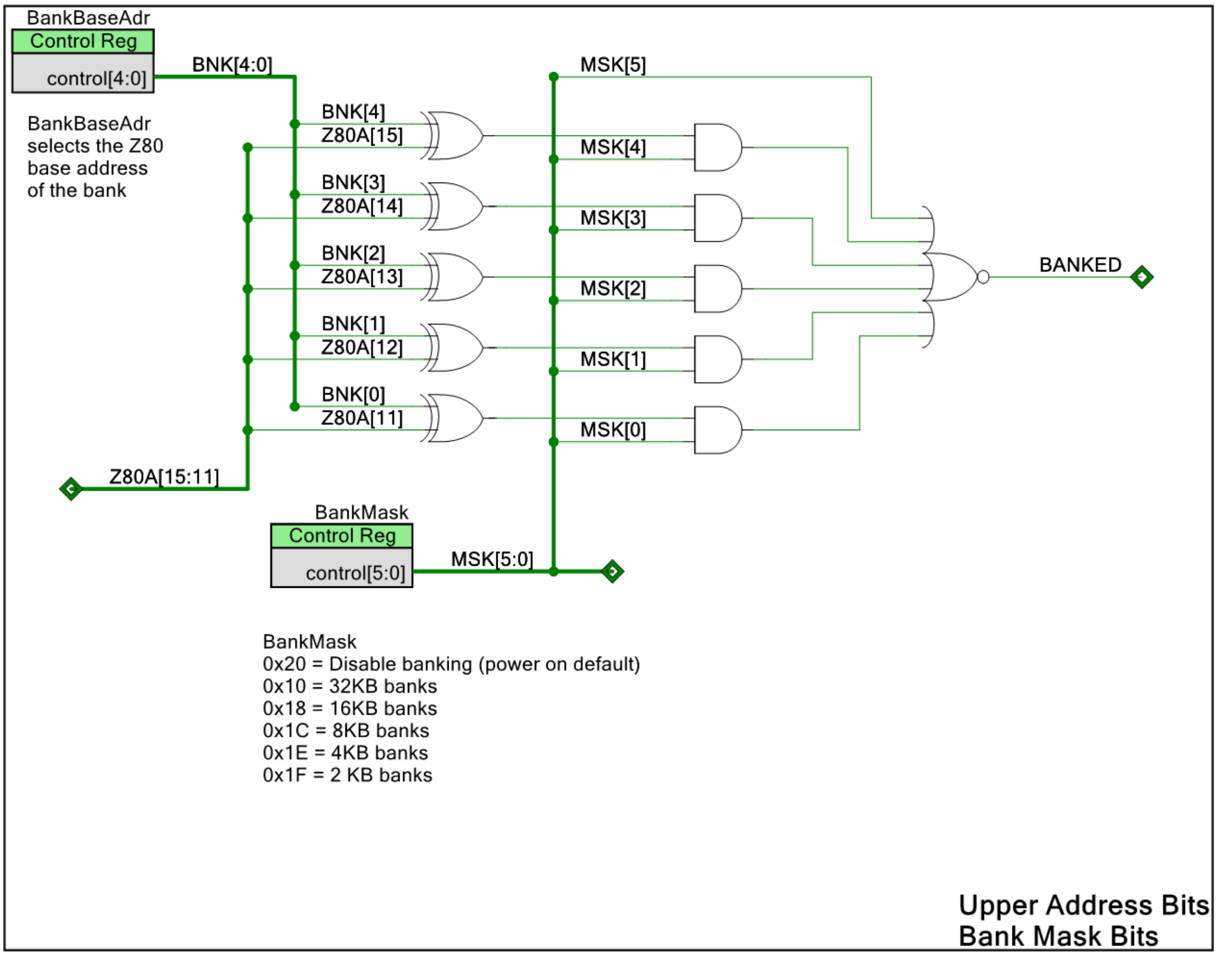


This schematic sheet contains the following component instances:

- Instance MMU_Sel (type: CyControlReg_v1_80)
- Instance mux_1 (type: mux_v1_10)
- Instance mux_10 (type: mux_v1_10)
- Instance mux_11 (type: mux_v1_10)
- Instance mux_2 (type: mux_v1_10)
- Instance mux_3 (type: mux_v1_10)
- Instance mux_4 (type: mux_v1_10)
- Instance mux_5 (type: mux_v1_10)
- Instance mux_6 (type: mux_v1_10)

7.4 Schematic Sheet: BankComp

Figure 8. Schematic Sheet: BankComp

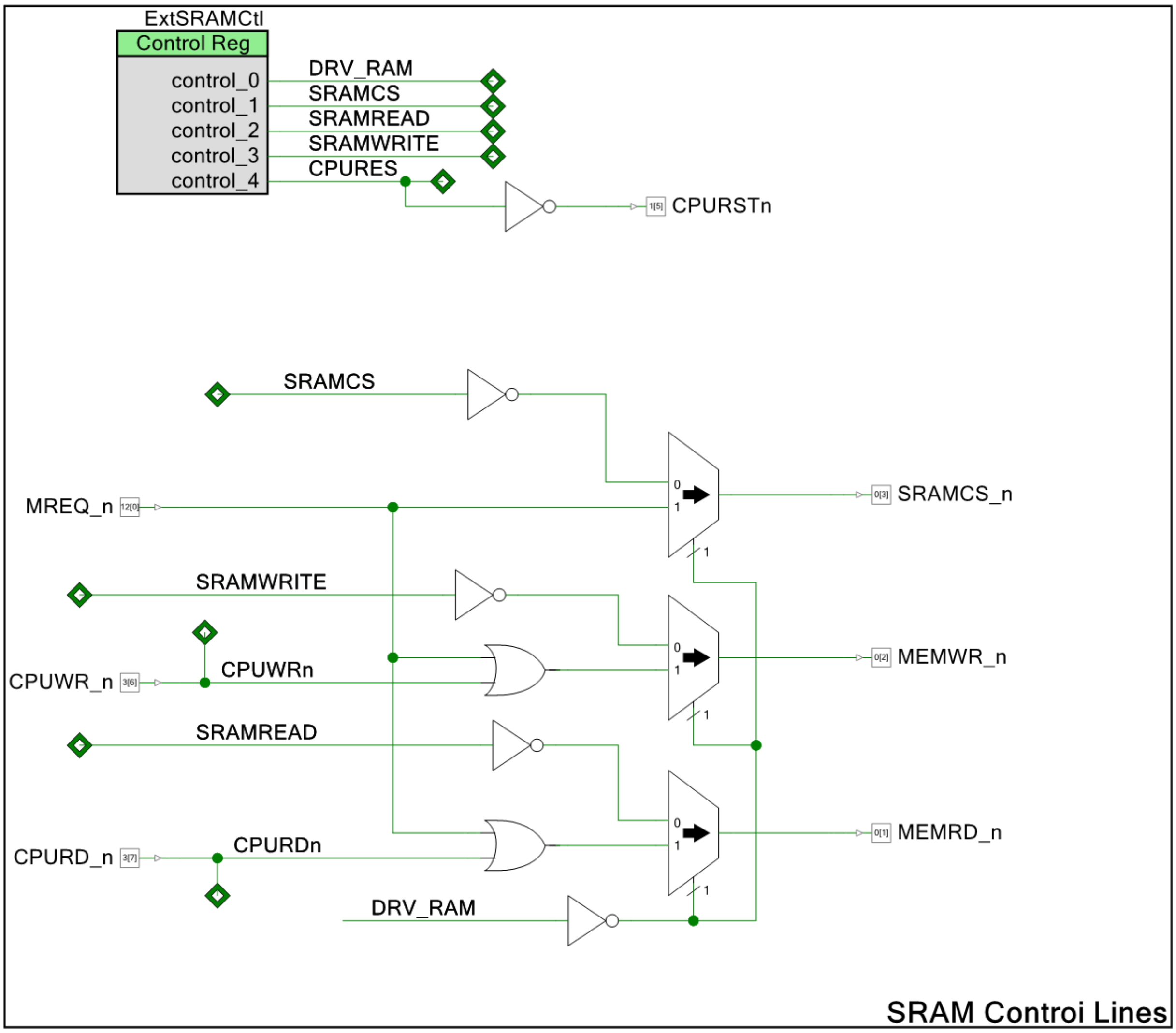


This schematic sheet contains the following component instances:

- Instance BankBaseAdr (type: CyControlReg_v1_80)
- Instance BankMask (type: CyControlReg_v1_80)

7.5 Schematic Sheet: SRAMCtlr

Figure 9. Schematic Sheet: SRAMCtlr

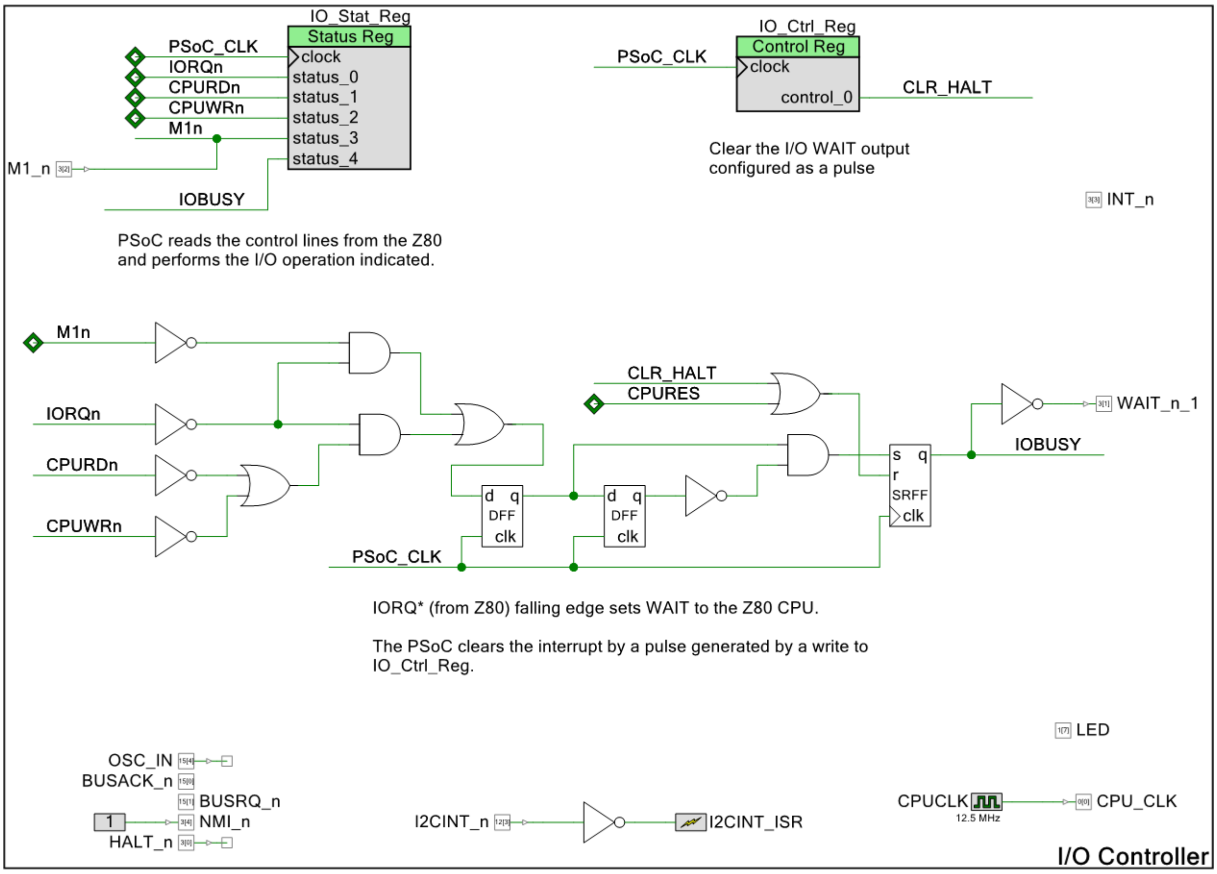


This schematic sheet contains the following component instances:

- Instance ExtSRAMCtl (type: CyControlReg_v1_80)
- Instance mux_7 (type: mux_v1_10)
- Instance mux_8 (type: mux_v1_10)
- Instance mux_9 (type: mux_v1_10)

7.6 Schematic Sheet: IO_Ctrlr

Figure 10. Schematic Sheet: IO_Ctrlr

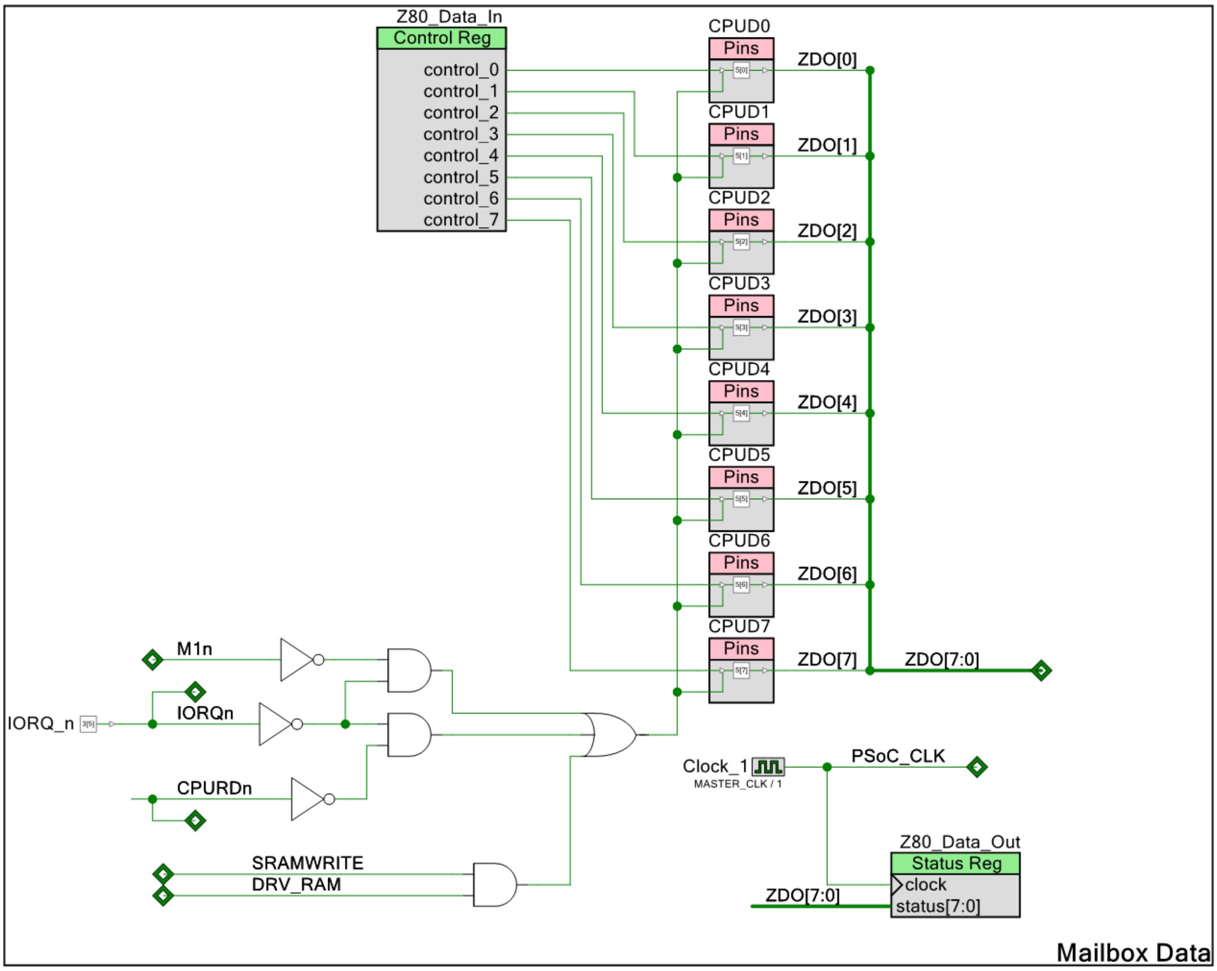


This schematic sheet contains the following component instances:

- Instance cy_srff_1 (type: cy_srff_v1_0)
- Instance cydff_10 (type: cydff_v1_30)
- Instance cydff_9 (type: cydff_v1_30)
- Instance IO_Ctrl_Reg (type: CyControlReg_v1_80)
- Instance IO_Stat_Reg (type: CyStatusReg_v1_90)

7.7 Schematic Sheet: MailboxData

Figure 11. Schematic Sheet: MailboxData

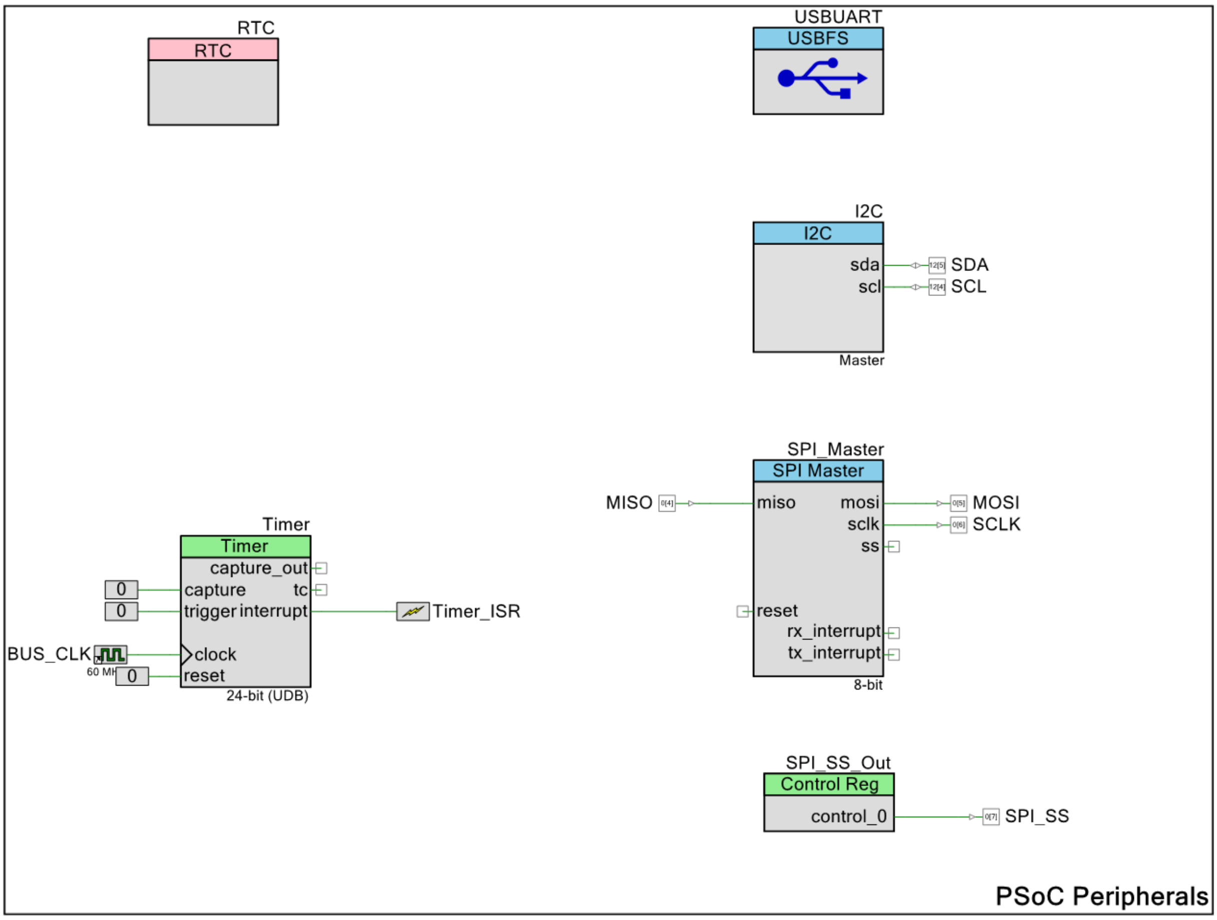


This schematic sheet contains the following component instances:

- Instance Z80_Data_In (type: CyControlReg_v1_80)
- Instance Z80_Data_Out (type: CyStatusReg_v1_90)

7.8 Schematic Sheet: Periphs

Figure 12. Schematic Sheet: Periphs



This schematic sheet contains the following component instances:

- Instance I2C (type: I2C_v3_50)
- Instance RTC (type: RTC_v2_0)
- Instance SPI_Master (type: SPI_Master_v2_50)
- Instance SPI_SS_Out (type: CyControlReg_v1_80)
- Instance Timer (type: Timer_v2_80)
- Instance USBUART (type: USBFS_v3_20)

7.9 Schematic Sheet: Sound

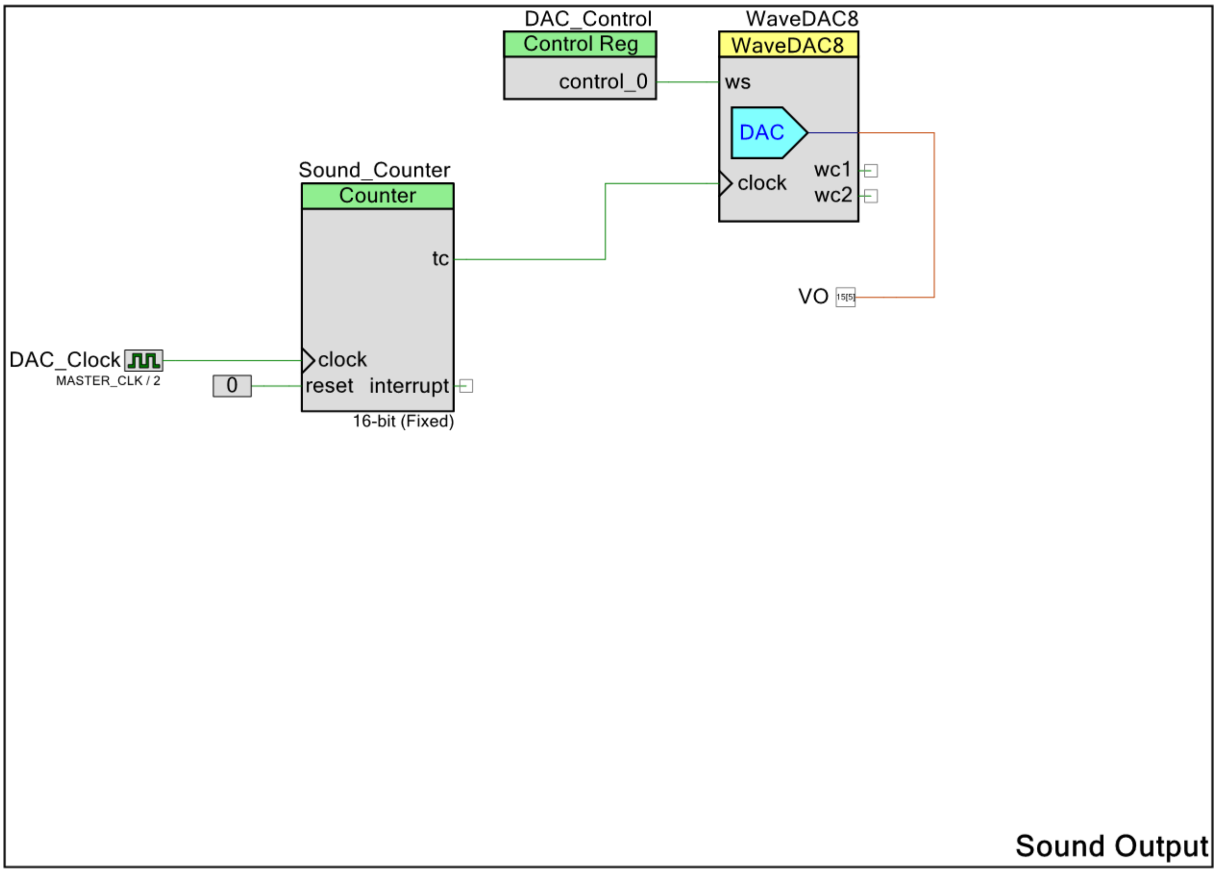


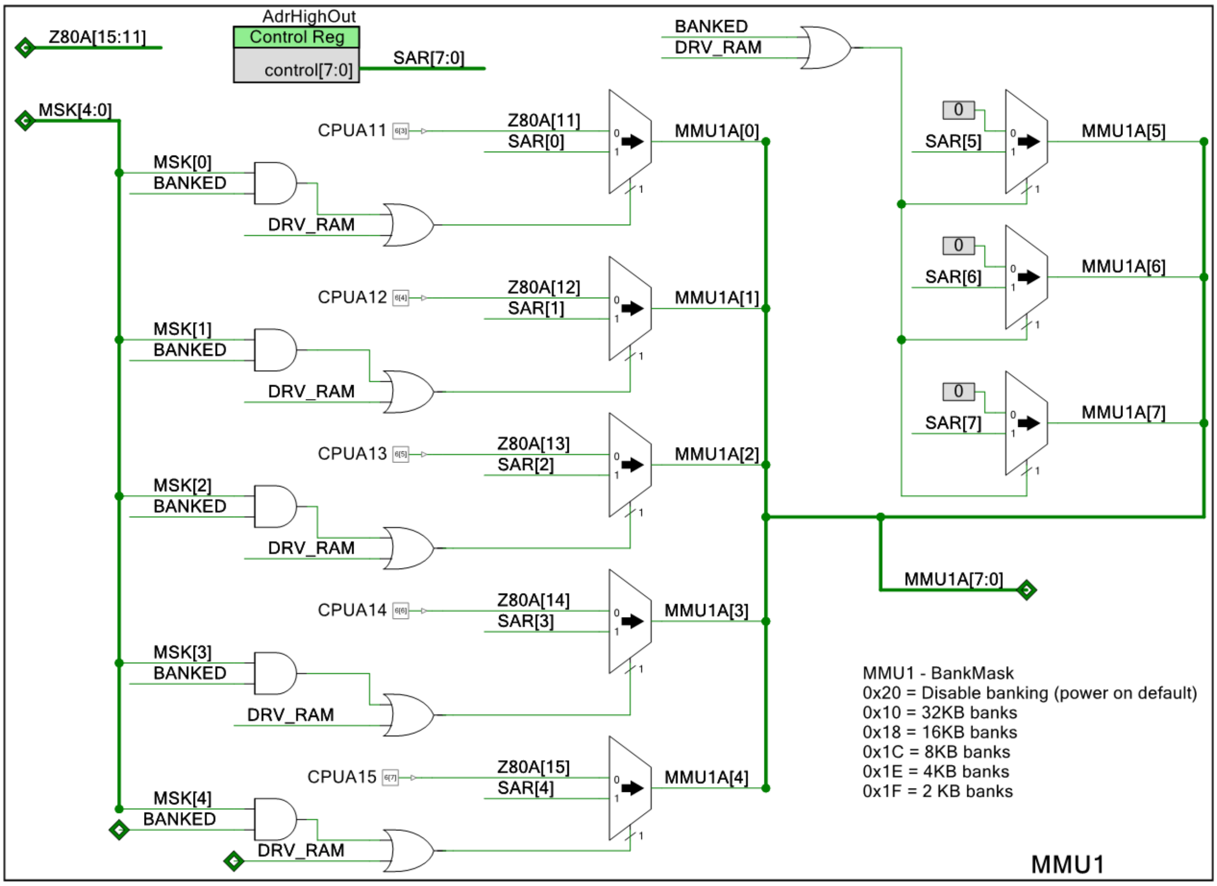
Figure 13. Schematic Sheet: Sound

This schematic sheet contains the following component instances:

- Instance DAC_Control (type: CyControlReg_v1_80)
- Instance Sound_Counter (type: Counter_v3_0)
- Instance WaveDAC8 (type: WaveDAC8_v2_10)

7.10 Schematic Sheet: MMU1

Figure 14. Schematic Sheet: MMU1

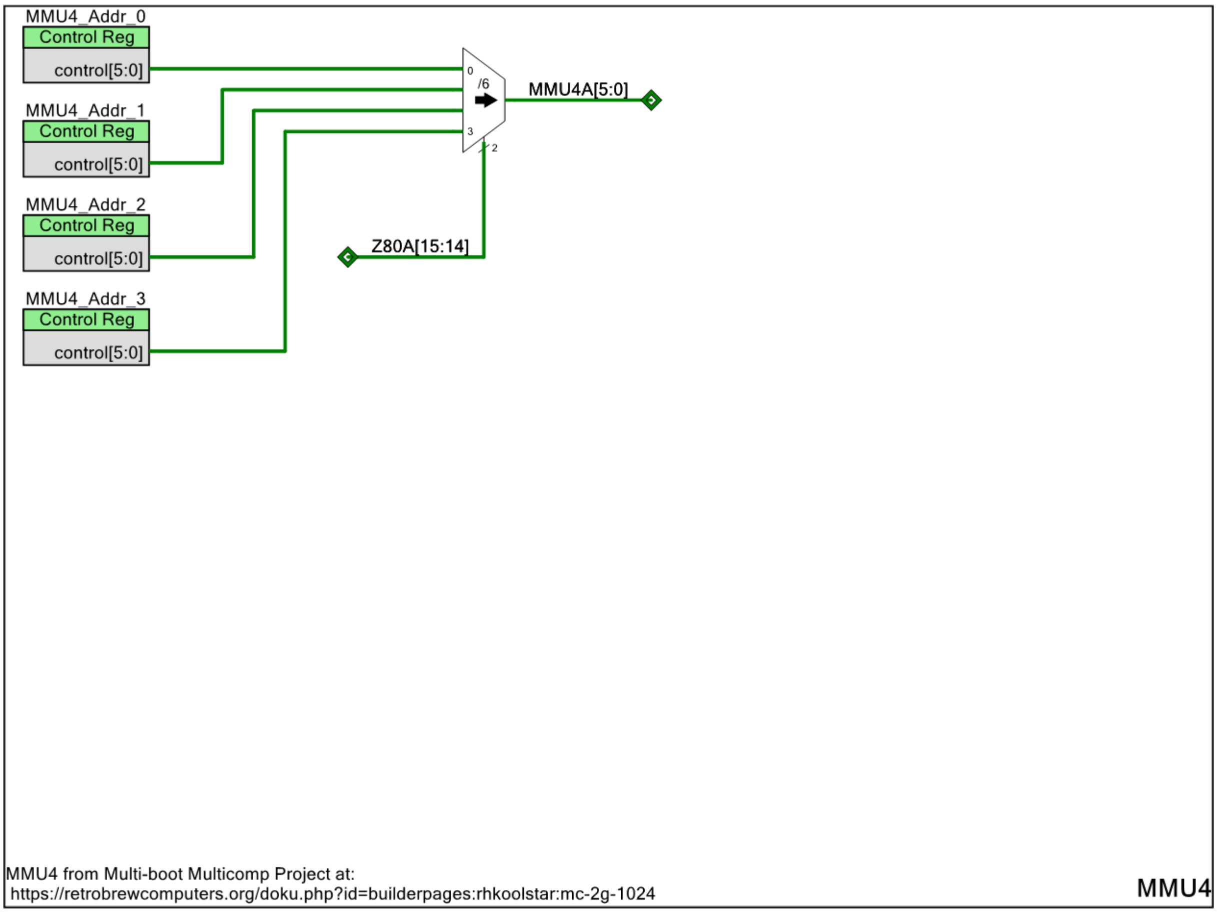


This schematic sheet contains the following component instances:

- Instance AdrHighOut (type: CyControlReg_v1_80)
- Instance mux_13 (type: mux_v1_10)
- Instance mux_14 (type: mux_v1_10)
- Instance mux_15 (type: mux_v1_10)
- Instance mux_16 (type: mux_v1_10)
- Instance mux_17 (type: mux_v1_10)
- Instance mux_18 (type: mux_v1_10)
- Instance mux_19 (type: mux_v1_10)
- Instance mux_20 (type: mux_v1_10)

7.11 Schematic Sheet: MMU4

Figure 15. Schematic Sheet: MMU4



This schematic sheet contains the following component instances:

- Instance MMU4_Addr_0 (type: CyControlReg_v1_80)
- Instance MMU4_Addr_1 (type: CyControlReg_v1_80)
- Instance MMU4_Addr_2 (type: CyControlReg_v1_80)
- Instance MMU4_Addr_3 (type: CyControlReg_v1_80)
- Instance mux_12 (type: mux_v1_10)

8 Components

8.1 Component type: Counter [v3.0]

8.1.1 Instance Sound_Counter

Description: 8, 16, 24 or 32-bit Counter
Instance type: Counter [v3.0]
Datasheet: online component datasheet for Counter

Table 14. Component Parameters for Sound_Counter

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the capture input. Default is None which does not have a capture input pin
ClockMode	Down Counter	Defines the operation of the counter. \nBasic: Count is incremented on the rising edge of the clock input. \n Clock_-And_Direction: Clock is incremented or decremented on the rising edge of the clock input based on the direction of the input. \nClock_And_UpCnt_-DwnCnt: Clock is an oversampling clock. On the rising edge of UpCnt, the counter is incremented and on the rising edge of DwnCnt, the counter is decremented.
CompareMode	Equal To	Specifies the compare output mode.
CompareStatusEdgeSense	true	Specifies whether rising edge sense for interrupt generation with the Compare output will be used. May be disabled to reduce resource usage.
CompareValue	65535	Defines the compare value. Valid vales are from 0 to the period value.
EnableMode	Software Only	Choose which enable controls the enable of the counter. This can be either through software with the control register, through hardware with the input pin or a combination of both where both must be active for the counter to be enabled.
FixedFunction	true	Defines whether Fixed Function Block usage is required.
InterruptOnCapture	false	Enables the counter status register to produce an interrupt output signal on a capture event.

Parameter Name	Value	Description
InterruptOnCompare	false	Enables the counter status register to produce an interrupt output signal on compare true.
InterruptOnOverUnderFlow	false	Enables the counter status register to produce an interrupt output signal on over flow or under flow.
InterruptOnTC	false	Enables the counter status register to produce an interrupt output signal on terminal count.
Period	65535	Defines the counter period value in clock counts from 1 to 2^Width-1.
ReloadOnCapture	false	Reloads the counter value to a set value on a capture input event.
ReloadOnCompare	false	Reloads the counter value to a set value on a compare equal event.
ReloadOnOverUnder	true	Reloads the counter value to a set value when overflow or underflow is detected.
ReloadOnReset	true	Reloads the counter value to a set value when reset input is high.
Resolution	16	Defines the width of the counter. It can be 8, 16, 24 or 32 (24 or 32 cannot use Fixed Function block).
RunMode	Continuous	Define the hardware operation to run continuously or run till a terminal count.
UseInterrupt	true	Allows for complete optimization of resource usage down to removing the status register if not required by the user.
User Comments		Instance-specific comments.

8.2 Component type: cy_srff [v1.0]

8.2.1 Instance cy_srff_1

Description: SR Flip Flop
Instance type: cy_srff [v1.0]
Datasheet: online component datasheet for cy_srff

Table 15. Component Parameters for cy_srff_1

Parameter Name	Value	Description
ArrayWidth	1	Width of s, r, and q terminals. Must be between 1 and 32.
User Comments		Instance-specific comments.

8.3 Component type: CyControlReg [v1.80]

8.3.1 Instance AdrHighOut

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 16. Component Parameters for AdrHighOut

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	8	Defines the number of outputs needed (1-8)
User Comments	Generates upper address bits for bank switching and loading the SRAM from the PSoC.	Instance-specific comments.

8.3.2 Instance AdrLowOut

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 17. Component Parameters for AdrLowOut

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	8	Defines the number of outputs needed (1-8)
User Comments	Address bits 0-7. Used by the CPU when it is loading the SRAM with the program.	Instance-specific comments.

8.3.3 Instance AdrMidOut

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 18. Component Parameters for AdrMidOut

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	3	Defines the number of outputs needed (1-8)
User Comments	The middle 3 bits of address are only driven by the PSoC when loading SRAM.	Instance-specific comments.

8.3.4 Instance BankBaseAdr

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 19. Component Parameters for BankBaseAdr

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	5	Defines the number of outputs needed (1-8)

Parameter Name	Value	Description
User Comments	Sets the base address of the bank switching logic.	Instance-specific comments.

8.3.5 Instance BankMask

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 20. Component Parameters for BankMask

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	32	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	6	Defines the number of outputs needed (1-8)
User Comments	Sets the size of the SRAM bank for bank switching.	Instance-specific comments.

8.3.6 Instance DAC_Control

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 21. Component Parameters for DAC_Control

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value

Parameter Name	Value	Description
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.7 Instance ExtSRAMCtl

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 22. Component Parameters for ExtSRAMCtl

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	16	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	5	Defines the number of outputs needed (1-8)
User Comments	Drives signals while loading the SRAM from the PSoC. D0 - DRVRAM - high to drive the SRAM from the PSoC. D1 - SRAMCS - Driven to create SRAM Chip Select during SRAM writes from the PSoC. D2 - SRAMREAD - Drive to read the SRAM from the PSoC. D3 - SRAMWRITE - Drive to read the SRAM from the PSoC. D4 - CPU Reset - Drive while loading the SRAM from the PSoC. Holds Z80 in reset which tri-states it's busses.	Instance-specific comments.

8.3.8 Instance IO_Ctrl_Reg

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 23. Component Parameters for IO_Ctrl_Reg

Parameter Name	Value	Description
Bit0Mode	PulseMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments	D0 - clears the HALT* line to the Z80. Done at the end of an IO cycle. D1 - On-boarrd LED. D2 - Set interrupt to the Z80. Cleared by Interrupt Acknowledgement cycle.	Instance-specific comments.

8.3.9 Instance MMU_Sel

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 24. Component Parameters for MMU_Sel

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.10 Instance MMU4_Addr_0

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 25. Component Parameters for MMU4_Addr_0

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	6	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.11 Instance MMU4_Addr_1

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 26. Component Parameters for MMU4_Addr_1

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	6	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.12 Instance MMU4_Addr_2

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 27. Component Parameters for MMU4_Addr_2

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode

Parameter Name	Value	Description
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	6	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.13 Instance MMU4_Addr_3

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 28. Component Parameters for MMU4_Addr_3

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	true	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	6	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.14 Instance SPI_SS_Out

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 29. Component Parameters for SPI_SS_Out

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode

Parameter Name	Value	Description
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.3.15 Instance Z80_Data_In

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 30. Component Parameters for Z80_Data_In

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	8	Defines the number of outputs needed (1-8)
User Comments	Data output to the Z80 from the PSoC. Driven for SRAM Writes during loading and for responding to I/O reads plus Interrupt Acknowledgement cycles.	Instance-specific comments.

8.4 Component type: cydff [v1.30]

8.4.1 Instance cydff_1

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 31. Component Parameters for cydff_1

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.

Parameter Name	Value	Description
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.2 Instance cydff_10

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 32. Component Parameters for cydff_10

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.3 Instance cydff_2

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 33. Component Parameters for cydff_2

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.4 Instance cydff_3

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 34. Component Parameters for cydff_3

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.

Parameter Name	Value	Description
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.5 Instance cydff_4

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 35. Component Parameters for cydff_4

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.6 Instance cydff_5

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 36. Component Parameters for cydff_5

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.7 Instance cydff_6

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 37. Component Parameters for cydff_6

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.

Parameter Name	Value	Description
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.8 Instance cydff_7

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 38. Component Parameters for cydff_7

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.9 Instance cydff_8

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 39. Component Parameters for cydff_8

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.4.10 Instance cydff_9

Description: D Flip Flop with configurable reset and preset
Instance type: cydff [v1.30]
Datasheet: online component datasheet for cydff

Table 40. Component Parameters for cydff_9

Parameter Name	Value	Description
ArrayWidth	1	Width of d and q terminals. Must be between 1 and 32.

Parameter Name	Value	Description
MultiPresetReset	true	Defines options to set the preset and reset to be either a bus or a single bit.
PresetOrReset	None	Preset/ Reset parameter
SmallMode	true	
User Comments		Instance-specific comments.

8.5 Component type: CyStatusReg [v1.90]

8.5.1 Instance AdrLowIn

Description: The Status Register allows the firmware to read values from digital signals.
Instance type: CyStatusReg [v1.90]
Datasheet: online component datasheet for CyStatusReg

Table 41. Component Parameters for AdrLowIn

Parameter Name	Value	Description
Bit0Mode	Transparent	Bit Mode for Bit 0 of the Status Register
Bit1Mode	Transparent	Bit Mode for Bit 1 of the Status Register
Bit2Mode	Transparent	Bit Mode for Bit 2 of the Status Register
Bit3Mode	Transparent	Bit Mode for Bit 3 of the Status Register
Bit4Mode	Transparent	Bit Mode for Bit 4 of the Status Register
Bit5Mode	Transparent	Bit Mode for Bit 5 of the Status Register
Bit6Mode	Transparent	Bit Mode for Bit 6 of the Status Register
Bit7Mode	Transparent	Bit Mode for Bit 7 of the Status Register
BusDisplay	false	Displays the input terminals as bus
Interrupt	false	Shows the interrupt terminal
MaskValue	0	Defines the value of the interrupt mask
NumInputs	8	Defines the number of status inputs (1-8)
User Comments	Latches the Z80 address. Set by I/O transfers and Interrupt Acknowledgement cycles. Used by the PSoC to determine the I/O address being accessed by the PSoC.	Instance-specific comments.

8.5.2 Instance IO_Stat_Reg

Description: The Status Register allows the firmware to read values from digital signals.
Instance type: CyStatusReg [v1.90]
Datasheet: online component datasheet for CyStatusReg

Table 42. Component Parameters for IO_Stat_Reg

Parameter Name	Value	Description
Bit0Mode	Transparent	Bit Mode for Bit 0 of the Status Register
Bit1Mode	Transparent	Bit Mode for Bit 1 of the Status Register
Bit2Mode	Transparent	Bit Mode for Bit 2 of the Status Register
Bit3Mode	Transparent	Bit Mode for Bit 3 of the Status Register
Bit4Mode	Transparent	Bit Mode for Bit 4 of the Status Register
Bit5Mode	Transparent	Bit Mode for Bit 5 of the Status Register
Bit6Mode	Transparent	Bit Mode for Bit 6 of the Status Register
Bit7Mode	Transparent	Bit Mode for Bit 7 of the Status Register
BusDisplay	false	Displays the input terminals as bus
Interrupt	false	Shows the interrupt terminal
MaskValue	0	Defines the value of the interrupt mask
NumInputs	5	Defines the number of status inputs (1-8)
User Comments	This register is polled by the PSoC to determine if there is an IO or interrupt acknowledge request present.	Instance-specific comments.

8.5.3 Instance Z80_Data_Out

Description: The Status Register allows the firmware to read values from digital signals.
Instance type: CyStatusReg [v1.90]
Datasheet: online component datasheet for CyStatusReg

Table 43. Component Parameters for Z80_Data_Out

Parameter Name	Value	Description
Bit0Mode	Transparent	Bit Mode for Bit 0 of the Status Register
Bit1Mode	Transparent	Bit Mode for Bit 1 of the Status Register
Bit2Mode	Transparent	Bit Mode for Bit 2 of the Status Register
Bit3Mode	Transparent	Bit Mode for Bit 3 of the Status Register
Bit4Mode	Transparent	Bit Mode for Bit 4 of the Status Register
Bit5Mode	Transparent	Bit Mode for Bit 5 of the Status Register
Bit6Mode	Transparent	Bit Mode for Bit 6 of the Status Register
Bit7Mode	Transparent	Bit Mode for Bit 7 of the Status Register
BusDisplay	true	Displays the input terminals as bus

Parameter Name	Value	Description
Interrupt	false	Shows the interrupt terminal
MaskValue	0	Defines the value of the interrupt mask
NumInputs	8	Defines the number of status inputs (1-8)
User Comments	Buffers data from the Z80 and is also used to read data by the PSoC when the Z80 does writes to I/O.	Instance-specific comments.

8.6 Component type: I2C [v3.50]

8.6.1 Instance I2C

Description: Standard I2C communication interface
Instance type: I2C [v3.50]
Datasheet: online component datasheet for I2C

Table 44. Component Parameters for I2C

Parameter Name	Value	Description
Address_Decode	Hardware	Determines either hardware or software address match logic.
BusSpeed_kHz	400	I2C Data Rate in kbps. Standard settings are 50, 100, 400 or 1000. The value must be between 1 and 1000.
EnableWakeup	false	Determines if I2C is selected as wakeup source.
ExternalBuffer	false	Exposes scl and sda in and out terminals outside the component.
Externi2cIntrHandler	false	Allows I2C interrupt handler to be set outside the I2C component. This feature intended only for PM/SM bus usage.
ExternTmoutIntrHandler	false	Allows I2C timeout interrupt handler to be set outside the I2C component. This feature intended only for PM/SM bus usage.
Hex	false	Indicates that address has been input in hexadecimal format.
I2C_Mode	Master	Determines I2C mode (Slave/Master/Multi-Master/Multi-Master-Slave).
I2cBusPort	Any	Determines which I2C pins have been selected. Select I2C0/I2C1 and connect to corresponding pins to be able use I2C as wakeup source.
Implementation	FixedFunction	Determines either I2C implementation Fixed Function or UDB.

Parameter Name	Value	Description
NotSlaveClockMinusTolerance	25	Internal component clock negative tolerance value in Master, Multi-Master or Multi-Master-Slave mode.
NotSlaveClockPlusTolerance	5	Internal component clock positive tolerance value in Master, Multi-Master or Multi-Master-Slave mode.
PrescalerEnabled	false	Enables prescaler (7-bit counter) to expand timeout timer range.
PrescalerPeriod	3	Prescaler period of timeout timer.
SclTimeoutEnabled	false	Enables low time monitoring of scl line.
SdaTimeoutEnabled	false	Enables low time monitoring of sda line.
Slave_Address	8	7-bits I2C slave address.
SlaveClockMinusTolerance	5	Internal component clock negative tolerance value in Slave mode.
SlaveClockPlusTolerance	50	Internal component clock positive tolerance value in Slave mode.
TimeoutImplementation	UDB	Determines either timeout timer feature implementation as UDB or Fixed Function. The Fixed Function implementation only available for PSoC5LP.
TimeOutms	25	Determines maximum time allowed for scl or sda to be low state (in mS). The timeout timer generates interrupt after timeout expires.
TimeoutPeriodff	39999	Period of timeout timer (Fixed Function).
TimeoutPeriodUdb	39999	Period of timeout timer (UDB).
UdbInternalClock	true	Determines either internal or external clock source for I2C UDB.
UdbSlaveFixedPlacementEnable	false	Enables fixed placement for I2C UDB. Only available in slave mode.
User Comments	Connects to the Front Panel and the Optional MCP23017 part.	Instance-specific comments.

8.7 Component type: mux [v1.10]

8.7.1 Instance mux_1

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 45. Component Parameters for mux_1

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.2 Instance mux_10

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 46. Component Parameters for mux_10

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.3 Instance mux_11

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 47. Component Parameters for mux_11

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.4 Instance mux_12

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 48. Component Parameters for mux_12

Parameter Name	Value	Description
NumInputTerminals	4	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	6	Width of each terminal
User Comments		Instance-specific comments.

8.7.5 Instance mux_13

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]

Datasheet: online component datasheet for mux

Table 49. Component Parameters for mux_13

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.6 Instance mux_14

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 50. Component Parameters for mux_14

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.7 Instance mux_15

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 51. Component Parameters for mux_15

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.8 Instance mux_16

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 52. Component Parameters for mux_16

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.9 Instance mux_17

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 53. Component Parameters for mux_17

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.10 Instance mux_18

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 54. Component Parameters for mux_18

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.11 Instance mux_19

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 55. Component Parameters for mux_19

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.12 Instance mux_2

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 56. Component Parameters for mux_2

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.13 Instance mux_20

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 57. Component Parameters for mux_20

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.14 Instance mux_3

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 58. Component Parameters for mux_3

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.15 Instance mux_4

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 59. Component Parameters for mux_4

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.16 Instance mux_5

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]

Datasheet: online component datasheet for mux

Table 60. Component Parameters for mux_5

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.17 Instance mux_6

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 61. Component Parameters for mux_6

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.18 Instance mux_7

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 62. Component Parameters for mux_7

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.19 Instance mux_8

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 63. Component Parameters for mux_8

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.7.20 Instance mux_9

Description: Multiplexer with configurable number of input terminals and terminal width.
Instance type: mux [v1.10]
Datasheet: online component datasheet for mux

Table 64. Component Parameters for mux_9

Parameter Name	Value	Description
NumInputTerminals	2	Number of inputs required for the Multiplexer. Acceptable values are 2, 4, 8, and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.8 Component type: RTC [v2.0]

8.8.1 Instance RTC

Description: Real Time Clock
Instance type: RTC [v2.0]
Datasheet: online component datasheet for RTC

Table 65. Component Parameters for RTC

Parameter Name	Value	Description
DstEnable	true	Enable Data Saving Time
StartOfWeek	Sunday	Start of new week
User Comments		Instance-specific comments.

8.9 Component type: SPI_Master [v2.50]

8.9.1 Instance SPI_Master

Description: Serial Peripheral Interface Master
Instance type: SPI_Master [v2.50]
Datasheet: online component datasheet for SPI_Master

Table 66. Component Parameters for SPI_Master

Parameter Name	Value	Description
BidirectMode	false	Bidirectional mode setting
ClockInternal	true	Allow use of the internal clock and desired bit rate or an external clock source
DesiredBitRate	400000	Desired Bit Rate in bps
HighSpeedMode	false	Enables using of the High Speed Mode
InterruptOnByteComplete	false	Set Initial Interrupt Source to Enable Interrupt on Byte Transfer Complete
InterruptOnRXFull	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Full
InterruptOnRXNotEmpty	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Not Empty

Parameter Name	Value	Description
InterruptOnRXOverrun	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Overrun
InterruptOnSPIDone	false	Set Initial Interrupt Source to Enable Interrupt on SPI Done
InterruptOnSPIIdle	false	Set Initial Interrupt Source to Enable Interrupt on SPI Idle
InterruptOnTXEmpty	false	Set Initial Interrupt Source to Enable Interrupt on TX FIFO Empty
InterruptOnTXNotFull	false	Set Initial Interrupt Source to Enable Interrupt on TX FIFO Not Full
Mode	CPHA = 0, CPOL = 0	SPI mode defines the Clock Phase and Clock Polarity desired
NumberOfDataBits	8	Set the Number of Data bits 3-16
RxBufferSize	4	Defines the amount of RAM Set asside for the RX Buffer
ShiftDir	MSB First	Set the Shift Out Direction
TxBufferSize	4	Defines the amount of RAM Set asside for the TX Buffer
User Comments		Instance-specific comments.
UseRxInternalInterrupt	false	Defines whether Rx internal interrupt is used or not
UseTxInternalInterrupt	false	Defines whether Tx internal interrupt is used or not

8.10 Component type: Timer [v2.80]

8.10.1 Instance Timer

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.80]
Datasheet: online component datasheet for Timer

Table 67. Component Parameters for Timer

Parameter Name	Value	Description
CaptureAlternatingFall	false	Enables data capture on either edge but not until a valid falling edge is detected first.
CaptureAlternatingRise	false	Enables data capture on either edge but not until a valid rising edge is detected first.
CaptureCount	2	The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed.

Parameter Name	Value	Description
CaptureCounterEnabled	false	Enables the capture counter to count capture events (up to 127) before a capture is triggered.
CaptureMode	Software Controlled	This parameter defines the capture input signal requirements to trigger a valid capture event
EnableMode	Software Only	This parameter specifies the methods in enabling the component. Hardware mode makes the enable input pin visible. Software mode may reduce the resource usage if not enabled.
FixedFunction	false	Configures the component to use fixed function HW block instead of the UDB implementation.
InterruptOnCapture	false	Parameter to check whether interrupt on a capture event is enabled or disabled.
InterruptOnFIFOFull	false	Parameter to check whether interrupt on a FIFO Full event is enabled disabled.
InterruptOnTC	true	Parameter to check whether interrupt on a TC is enabled or disabled.
NumberOfCaptures	1	Number of captures allowed until the counter is cleared or disabled.
Period	1199999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	24	Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals.
RunMode	One Shot	Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered.
TriggerMode	Software Controlled	Defines the required trigger input signal to cause a valid trigger enable of the timer
User Comments		Instance-specific comments.

8.11 Component type: USBFS [v3.20]

8.11.1 Instance USBUART

Description: USB 2.0 Full Speed Device Framework
Instance type: USBFS [v3.20]
Datasheet: online component datasheet for USBFS

Table 68. Component Parameters for USBUART

Parameter Name	Value	Description
EnableBatteryChargDetect	false	This parameter allows to detect a charging supported USB host port using the API function USBFS_DetectPortType().
EnableCDCApi	true	Enables additional high level API's that allow the CDC device to be used similar to a UART device.
EnableMidiApi	true	Enables additional high level MIDI API's.
endpointMA	MA_Static	Endpoint memory allocation
endpointMM	EP_Manual	Endpoint memory management
epDMAautoOptimization	false	This parameter enables resource optimization for DMA with Automatic Memory Management mode. Set this parameter value to true only when a single IN endpoint is present in the device. Enabling this parameter in a multi IN endpoint device configuration causes undesired effects.
extern_cls	false	This parameter allows for user or other component to implement his own handler for Class requests. USBFS_-DispatchClassRqst() function should be implemented if this parameter enabled.
extern_vbus	false	This parameter enables external VBUSDET input.
extern_vnd	false	This parameter allows for user or other component to implement his own handler for Vendor specific requests. USBFS_HandleVendorRqst() function should be implemented if this parameter enabled.
extJackCount	0	Max number of External MIDI IN Jack or OUT Jack descriptors
Gen16bitEpAccessApi	true	This parameter defines whether to generate APIs for the 16-bits endpoint access.
HandleMscRequests	true	This parameter is used to enable handling MSC requests and generate MSC APIs.
isrGroupArbiter	High	This parameter defines the interrupt group of the Arbiter Interrupt.
isrGroupBusReset	Low	This parameter defines the interrupt group of the Bus Reset Interrupt.
isrGroupEp0	Medium	This parameter defines the interrupt group of the Control Endpoint Interrupt (EP0).

Parameter Name	Value	Description
isrGroupEp1	Medium	This parameter defines the interrupt group of the Data Endpoint 1 Interrupt.
isrGroupEp2	Medium	This parameter defines the interrupt group of the Data Endpoint 2 Interrupt.
isrGroupEp3	Medium	This parameter defines the interrupt group of the Data Endpoint 3 Interrupt.
isrGroupEp4	Medium	This parameter defines the interrupt group of the Data Endpoint 4 Interrupt.
isrGroupEp5	Medium	This parameter defines the interrupt group of the Data Endpoint 5 Interrupt.
isrGroupEp6	Medium	This parameter defines the interrupt group of the Data Endpoint 6 Interrupt.
isrGroupEp7	Medium	This parameter defines the interrupt group of the Data Endpoint 7 Interrupt.
isrGroupEp8	Medium	This parameter defines the interrupt group of the Data Endpoint 8 Interrupt.
isrGroupLpm	High	This parameter defines the interrupt group of the LPM Interrupt.
isrGroupSof	Low	This parameter defines the interrupt group of the Start of Frame Interrupt.
max_interfaces_num	2	Defines maximum interfaces number
Mode	false	Specifies whether the implementation will create API for interfacing to UART component(s) for a corresponding set of external MIDI connections.
mon_vbus	false	The mon_vbus parameter adds a single VBUS monitor pin to the design. This pin must be connected to VBUS and must be assigned in the pin editor.
MscDescriptors		Mass Storage Class Descriptors
MscLogicalUnitsNum	1	This parameter allows to specify the number of logical units that should be supported by the Mass Storage device.
out_sof	false	The out_sof parameter enables Start-of-Frame output.
Pid	F232	Product ID
powerpad_vbus	false	This parameter enables VBUS power pad
ProdactName		This string is displayed by the Operating System when it is installing the mass storage device as the Product Name.

Parameter Name	Value	Description
ProdactRevision		This string is displayed by the Operating System when it is installing the mass storage device as the Product Revision.
rm_lpm_int	true	Removes LPM ISR
User Comments	USB to Serial interface that connects the PSoC to the Host computer.	Instance-specific comments.
VendorName		This string is displayed by the Operating System when it is installing the mass storage device as the Vendor Name.
Vid	04B4	Vendor ID

8.12 Component type: WaveDAC8 [v2.10]

8.12.1 Instance WaveDAC8

Description: 8-Bit Waveform DAC
Instance type: WaveDAC8 [v2.10]
Datasheet: online component datasheet for WaveDAC8

Table 69. Component Parameters for WaveDAC8

Parameter Name	Value	Description
Clock_SRC	External	Select either internal or external clock source
DAC_Range	VDAC 0 - 1.020V	Define the type and range of the DAC
Sample_Clock_Freq	100000	Define the sample rate
User Comments		Instance-specific comments.
Wave1_Amplitude	1	Defines the peak-to-peak amplitude of the non-arbitrary waveform
Wave1_Data	128u, 135u, 143u, 151u, 159u, 166u, 174u, 181u, 188u, 194u, 201u, 207u, 213u, 219u, 224u, 229u, 233u, 237u, 241u, 244u, 246u, 249u, 250u, 252u, 252u, 252u, 252u, 252u, 250u, 249u, 246u, 244u, 241u, 237u, 233u, 229u, 224u, 219u, 213u, 207u, 201u, 194u, 188u, 181u, 174u, 166u, 159u, 151u, 143u, 135u, 128u, 120u, 112u, 104u, 96u, 89u, 81u, 74u, 67u, 61u, 54u, 48u, 42u, 36u, 31u, 26u, 22u, 18u, 14u, 11u, 9u, 6u, 5u, 3u, 3u, 3u, 3u, 3u, 5u, 6u, 9u, 11u, 14u, 18u, 22u, 26u, 31u, 36u, 42u, 48u, 54u, 61u, 67u, 74u, 81u, 89u, 96u, 104u, 112u, 120u	This is the storage data array for the content of the waveform
Wave1_DCOffset	0.51	Defines the non-arbitrary DC offset (relative to GND) of the waveform

Parameter Name	Value	Description
Wave1_Length	100	Defines the number of data points in the waveform
Wave1_PhaseShift	0	Defines the Phase Shift relative to the beginning of the waveform
Wave1_Type	Sine	Defines the waveform Type
Wave2_Amplitude	1	Defines the peak-to-peak amplitude of the non-arbitrary waveform
Wave2_Data	128u, 132u, 138u, 143u, 148u, 152u, 158u, 162u, 168u, 172u, 178u, 182u, 188u, 192u, 198u, 202u, 208u, 213u, 218u, 222u, 228u, 232u, 238u, 242u, 248u, 252u, 248u, 242u, 238u, 232u, 228u, 222u, 218u, 212u, 208u, 202u, 198u, 192u, 188u, 182u, 178u, 173u, 168u, 162u, 158u, 152u, 148u, 143u, 138u, 132u, 128u, 122u, 118u, 112u, 107u, 102u, 97u, 93u, 88u, 83u, 78u, 73u, 68u, 62u, 57u, 52u, 47u, 42u, 37u, 33u, 28u, 23u, 18u, 13u, 8u, 3u, 8u, 13u, 18u, 23u, 28u, 33u, 37u, 42u, 47u, 52u, 57u, 62u, 68u, 73u, 78u, 83u, 88u, 93u, 97u, 102u, 107u, 112u, 118u, 122u	This is the storage data array for the content of the waveform
Wave2_DCOffset	0.51	Defines the non-arbitrary DC offset (relative to GND) of the waveform
Wave2_Length	100	Defines the number of data points in the waveform
Wave2_PhaseShift	0	Defines the Phase Shift relative to the beginning of the waveform
Wave2_Type	Triangle	Defines the waveform Type

9 Other Resources

The following documents contain important information on Cypress software APIs that might be relevant to this design:

- Standard Types and Defines chapter in the System Reference Guide
  - Software base types
  - Hardware register types
  - Compiler defines
  - Cypress API return codes
  - Interrupt types and macros
- Registers
  - The full PSoC 5LP register map is covered in the PSoC 5LP Registers Technical Reference Manual
  - Register Access chapter in the System Reference Guide
    - CY_GET API routines
    - CY_SET API routines
- System Functions chapter in the System Reference Guide
  - General API routines
  - CyDelay API routines
  - CyVd Voltage Detect API routines
- Power Management
  - Power Supply and Monitoring chapter in the PSoC 5LP Technical Reference Manual
  - Low Power Modes chapter in the PSoC 5LP Technical Reference Manual
  - Power Management chapter in the System Reference Guide
    - CyPm API routines
- Watchdog Timer chapter in the System Reference Guide
  - CyWdt API routines
- Cache Management
  - Cache Controller chapter in the PSoC 5LP Technical Reference Manual
  - Cache chapter in the System Reference Guide
    - CyFlushCache() API routine