Datapath Configuration Tool Cheat Sheet

This document tells you all you need to know about the Datapath Configuration Tool to configure datapaths.

Contents

Datapath Configuration Tool Introduction 2
- Dynamic Configuration RAM (CFGRAM) Section 4
- Static Configuration Section 7
  - CFG9 Register 7
  - CFG11 and CFG10 Registers 8
  - CFG13 and CFG12 Registers 9
  - CFG15 and CFG14 Registers 10
  - CFG17 and CFG16 Registers 12
- Setting Initial Register Values 14
- Datapath Chaining 14
- Firmware-Control of Datapath Registers 16
- Miscellaneous 17

Datapath Configuration Tool Introduction

The Datapath is an 8-state state machine, while the Datapath Configuration Tool (DCT) is a bit-banger made easy through a GUI. This document shows the relationship between the DCT and the underlying Datapath hardware.

The GUI can be divided into two general sections – the Dynamic Configuration and Static Configuration sections, as Figure 1 shows.

- Dynamic Configuration – Allows you to set up the Datapath to behave differently across states
- Static Configuration – Stays the same across states

Figure 1. Datapath Configuration Tool Interface Sections

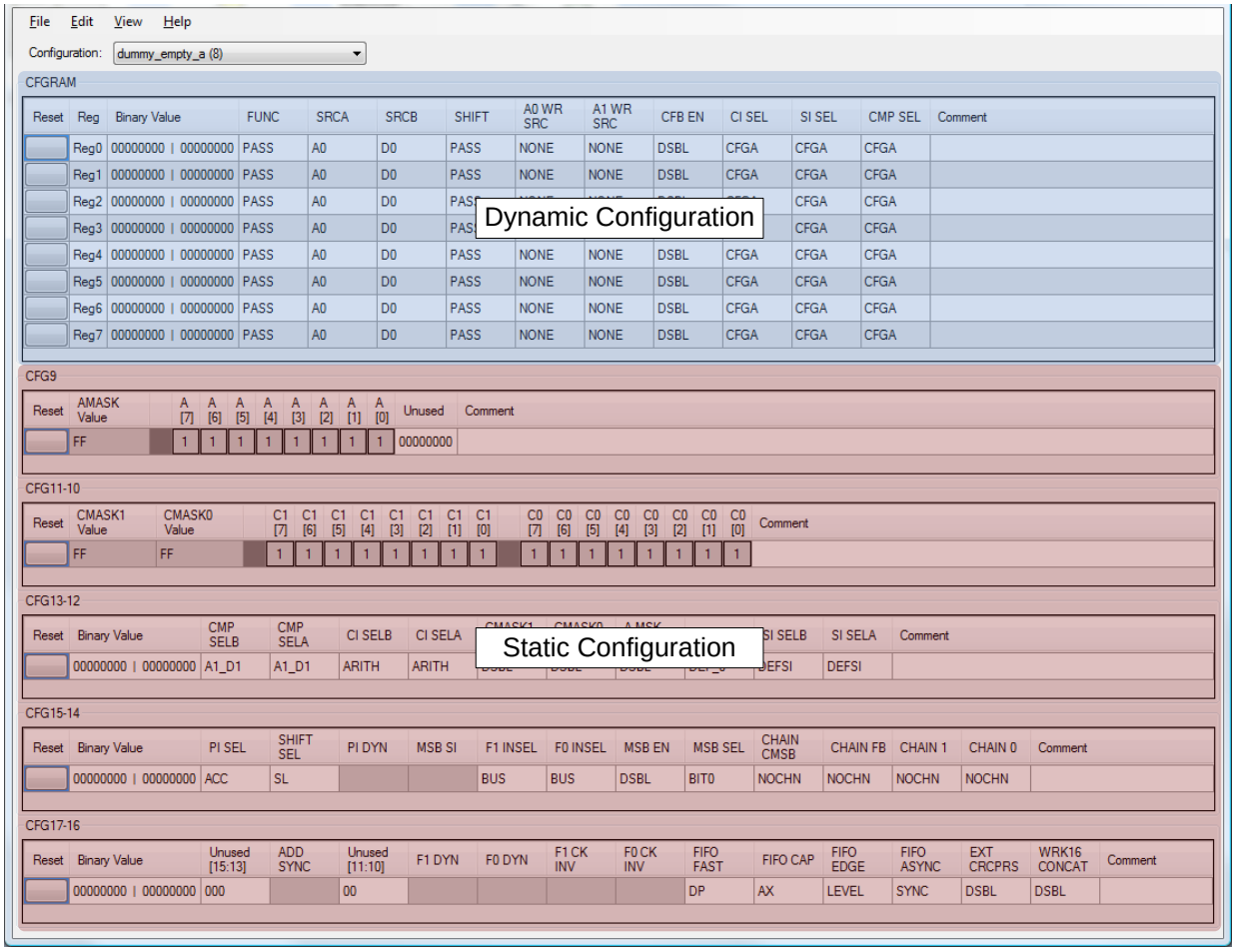
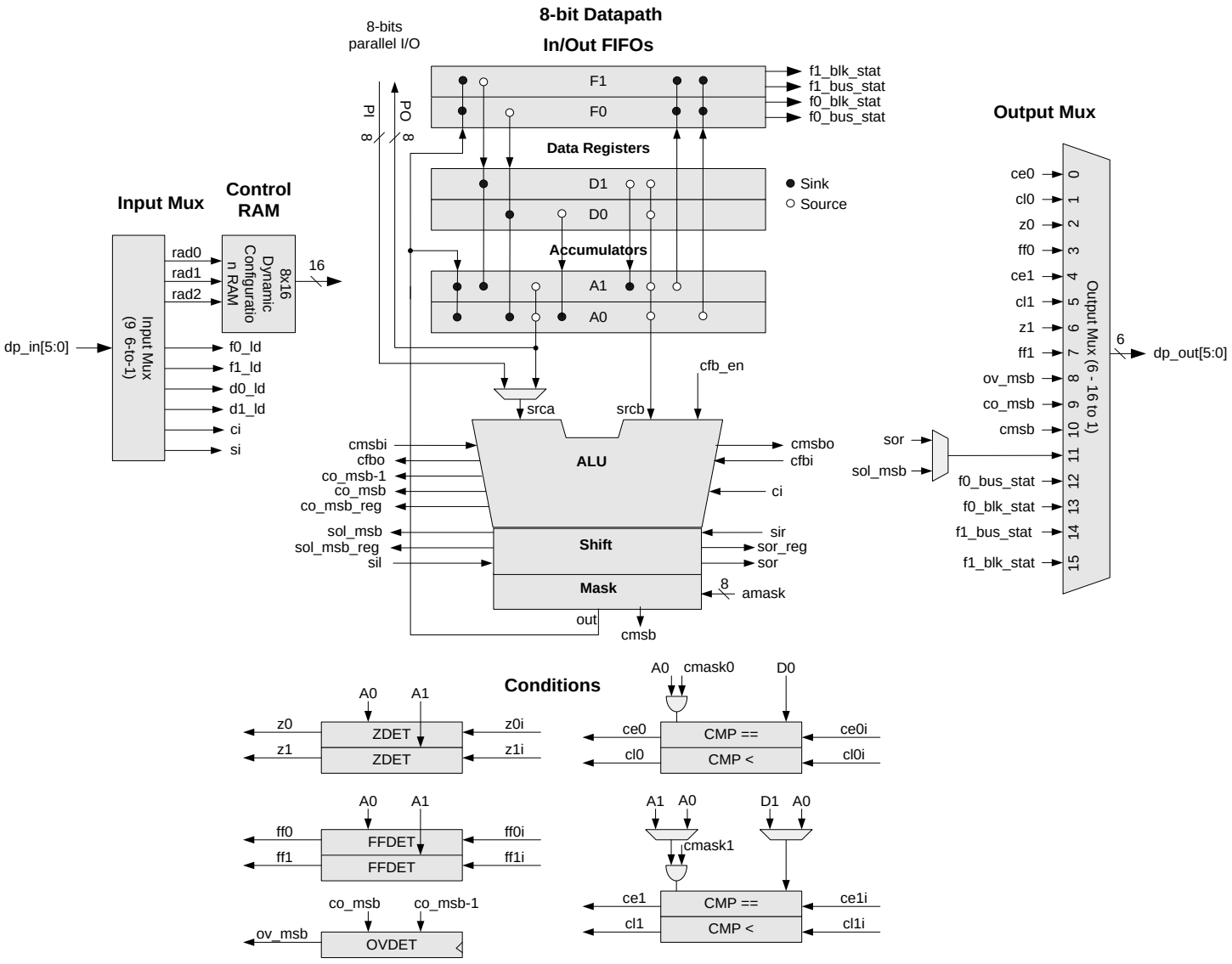


Figure 2. Datapath Block Diagram



Dynamic Configuration RAM (CFGRAM) Section

The Dynamic Configuration section is a representation of the configuration RAM. It configures the behavior of the datapath in the 8 ‘states’ of the state machine. The following tables explain the function of each of the fields in the GUI.

Table 1. The CFGRAM Section of the Datapath Configuration Tool

Dynamic Configuration RAM Section
Datapath Configuration Tool

CFGRAM

Reset	Reg	Binary Value	FUNC	SRCA	SRCB	SHIFT	A0 WR SRC	A1 WR SRC	CFB EN	CI SEL	SI SEL	CMP SEL
	Reg0	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg1	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg2	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg3	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg4	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg5	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg6	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA
	Reg7	00000000 | 00000000	PASS	A0	D0	PASS	NONE	NONE	DSBL	CFGA	CFGA	CFGA

Datapath Block Diagram

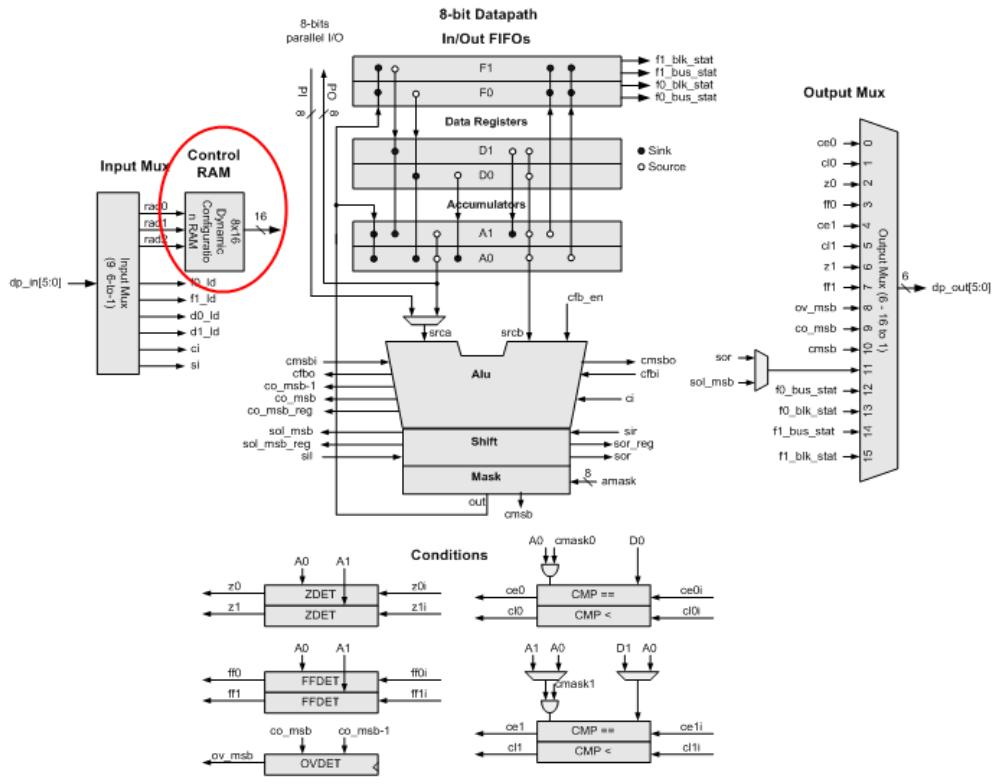


Table 2. Dynamic Configuration Section Column Descriptions

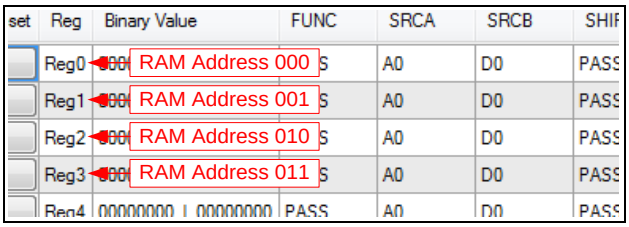
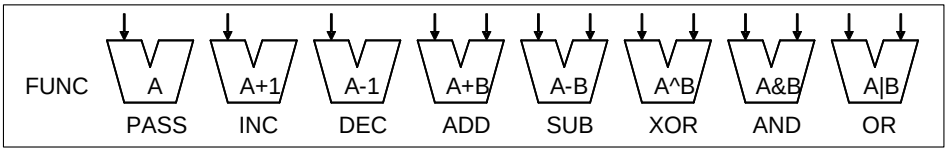
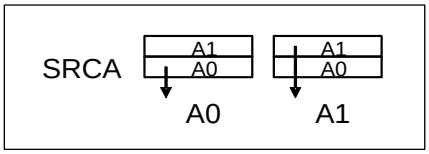
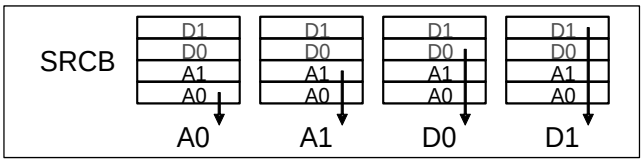
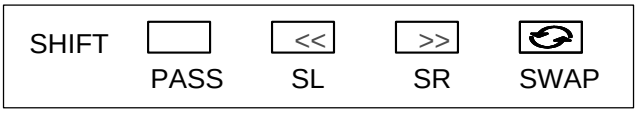
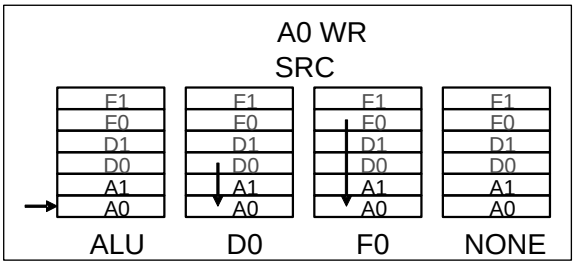
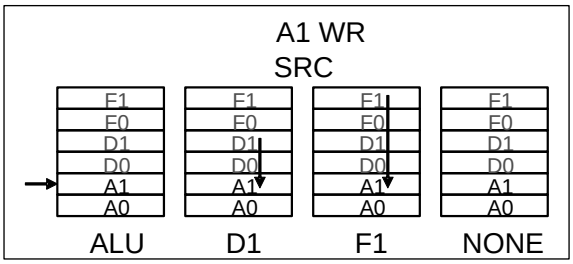
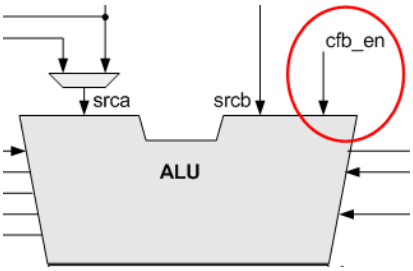
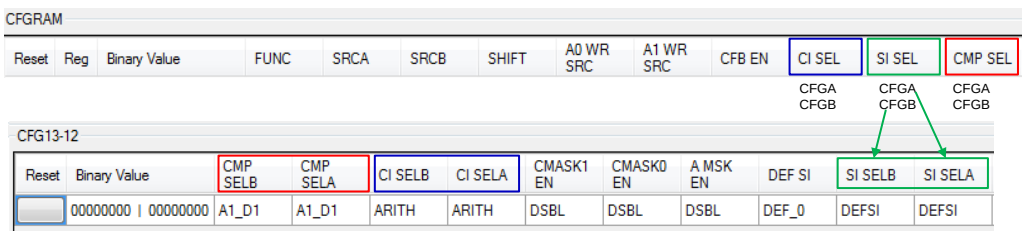
Dynamic Configuration Section	
Reg This column shows the 'state' to which the row corresponds. The value of the three CFGRAM address signals determines which configuration is selected, so take care to select the correct row for each configuration.	
FUNC This column determines which of the eight ALU functions will be performed in that configuration.	
SRCA This column determines the source for the ALU's 'srca' input. srca can also come from PI – see Table 6 (CFG 15-14).	
SRCB This column determines the source for the ALU's 'srcb' input.	
SHIFT This column determines the function of the shift block.	
A0 WR SRC This column determines the contents of the A0 register *after* the ALU operation is complete.	

Table 2. Dynamic Configuration Section Column Descriptions (contd.)

Dynamic Configuration Section	
A1 WR SRC This column determines the contents of the A1 register *after* the ALU operation is complete.	
CFB EN CRC config enable – see PI DYN in CFG15 and CFG14 Registers Table 6. CFG15 and CFG14 Register Definitons	
CI SEL, SI SEL, CMP SEL Carry in Select, Shift In Select, and Compare Select. Each of these has two configuration options: A and B. You can choose the one you want for each configuration. See Table 5 (CFG13-12) for more information.	

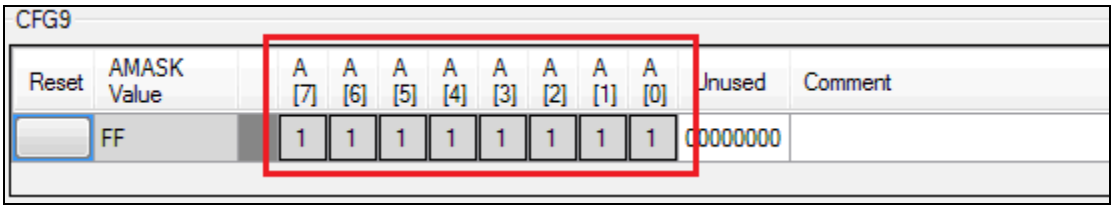
Static Configuration Section

The Static Configuration section represents the datapath registers CFG9 to CFG17, as Table 3, Table 4, Table 5, Table 6, and Table 7 show. They control static functions, including shift direction, masking, FIFO configuration, and chaining.

CFG9 Register

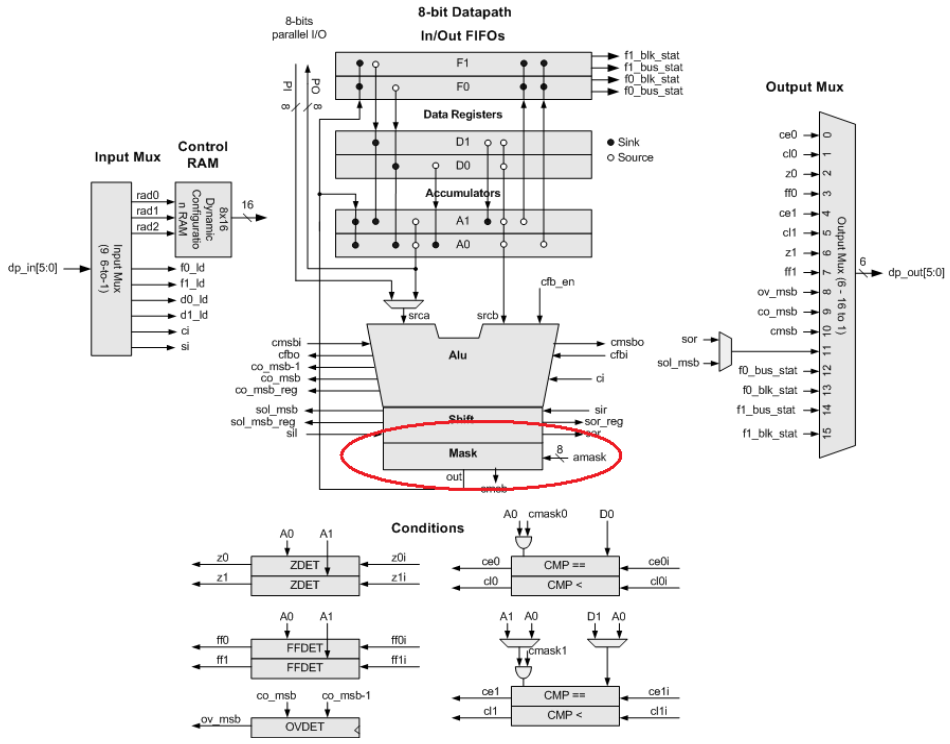
Table 3. CFG9 Register Definition

AMASK Value
Datapath Configuration Tool



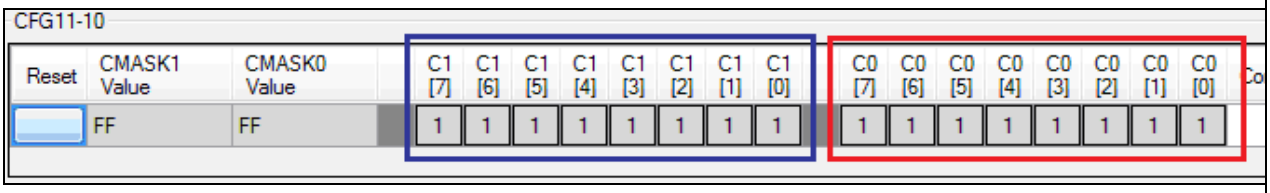
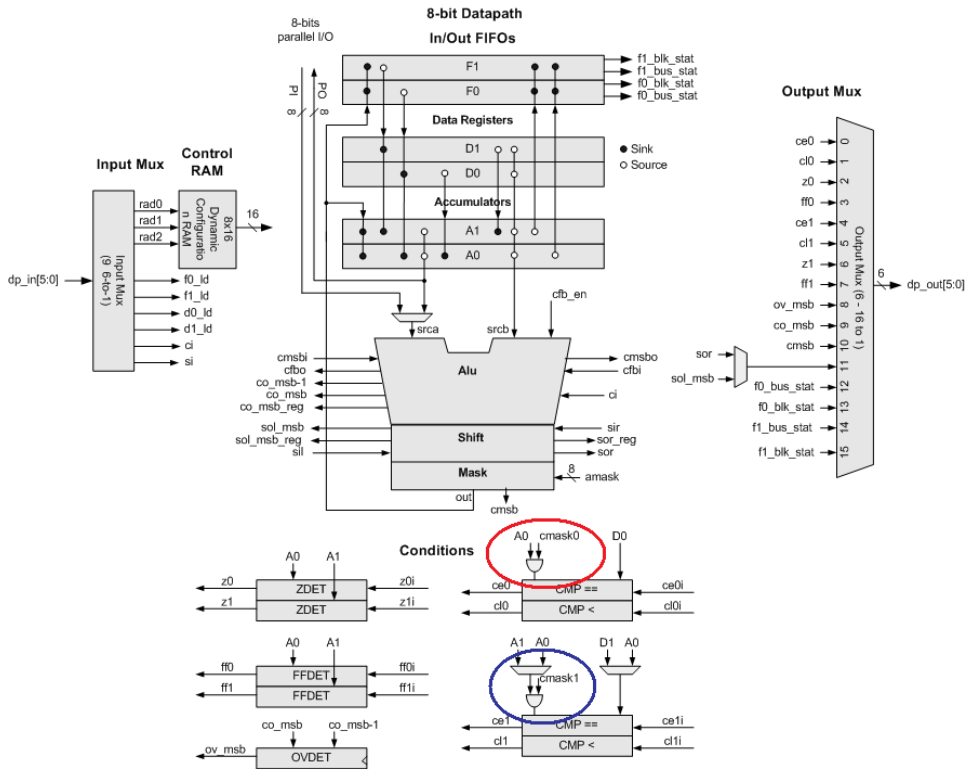
- AMASK Value – This field contains the 8-bit mask value that is applied to the output of the Datapath ALU block. The output of the shift register is ANDed with the contents of this register. This feature is off by default. To enable it the AMASK EN bit CFG12 must be set.

Datapath Block Diagram



CFG11 and CFG10 Registers

Table 4. CFG11 and CFG10 Register Definitions

CMASK0 and CMASK1 Values
Datapath Configuration Tool

■ CMASK0 and CMASK1 – These fields set the mask values used with the comparator block inputs. The contents of the A0 or A1 register are ANDed with the contents of these registers before comparison. To enable them, the CMASK0 EN and CMASK1 EN bits must be set in CFG12 (Table 5).
Datapath Block Diagram


CFG13 and CFG12 Registers

Table 5. CFG13 and CFG12 Register Definitions

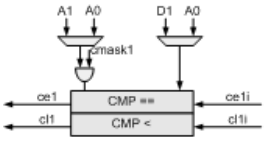
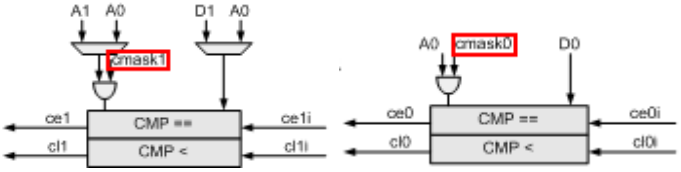
CFG13-12 Configuration Selections

Datapath Configuration Tool

CFG13-12

Reset	Binary Value	CMP SELB	CMP SELA	CI SELB	CI SELA	CMASK1 EN	CMASK0 EN	A MSK EN	DEF SI	SI SELB	SI SELA
	00000000 | 00000000	A1_D1	A1_D1	ARITH	ARITH	DSBL	DSBL	DSBL	DEF_0	DEFSI	DEFSI

CFG13-12 Details

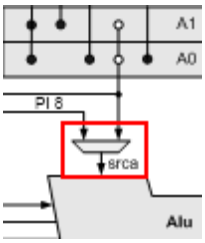
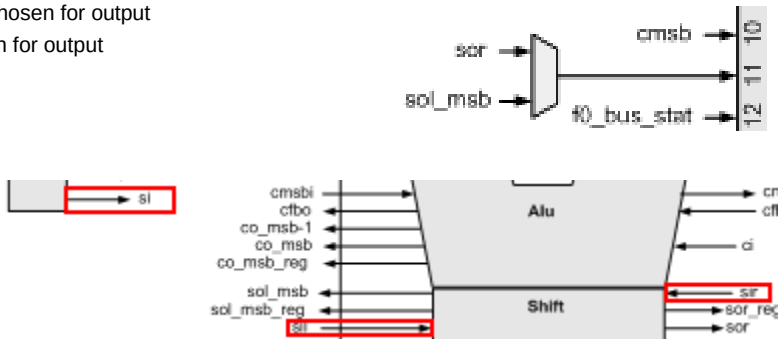
CMP SELB & CMP SEL A Configures the Comparison B and Comparison A Options for compare block 1. The CMP SEL field of the RAM configuration determines if the A or B option is in effect for a given cycle. Note: Compare block 0 can compare only D0 and a masked value from A0.	**A1_D1**: A1 < D1, A1 == D1 **A1_A0**: A1 < A0, A1 == A0 **A0_D1**: A0 < D1, A0 == D1 **A0_A0**: A0 < A0 , A0 == A0 
CI SELB & CI SEL A Selects the source of the carry in.	**ARITH:** The carry is controlled by ALU arithmetic. **REGIS:** The carry in is carry out registered from previous cycle. **ROUTE:** Carry in is selected from one of the datapath inputs. **CHAIN:** Carry in is driven from previous datapath in chain.
CMASK0 EN, CMASK1 EN, AMASK EN Enables the Masks on the Compare 1 and Compare 0 Blocks (shown right) and the Mask block at the output of the ALU (not shown). See Table 3 and Table 4.	
DEF SI Defines whether the default shift in value is a 0 or a 1. Note SI SELB or SI SELA must be set to DEFSI for this to matter.	**DEF_0:** Zero **DEF_1:** One
SI SELB & SI SEL A Selects the source of the shift in for the A and B shift in configuration.	**DEFSI:** Shifts in either a zero or a one, as defined by DEF SI. **REGIS:** Shift in is shift out register from previous cycle. **ROUTE:** Shift in is selected from a datapath input. **CHAIN:** Shift in is driven by previous datapath in chain.

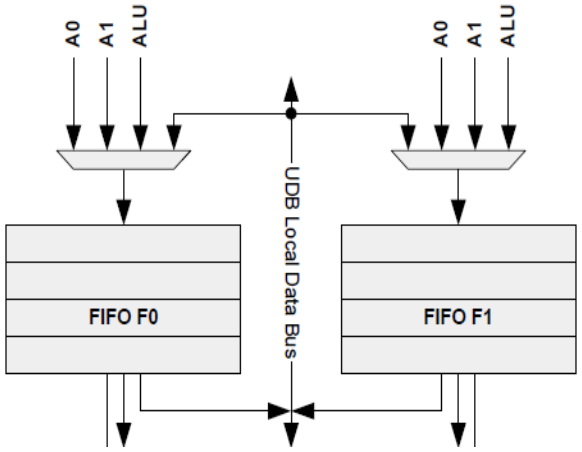
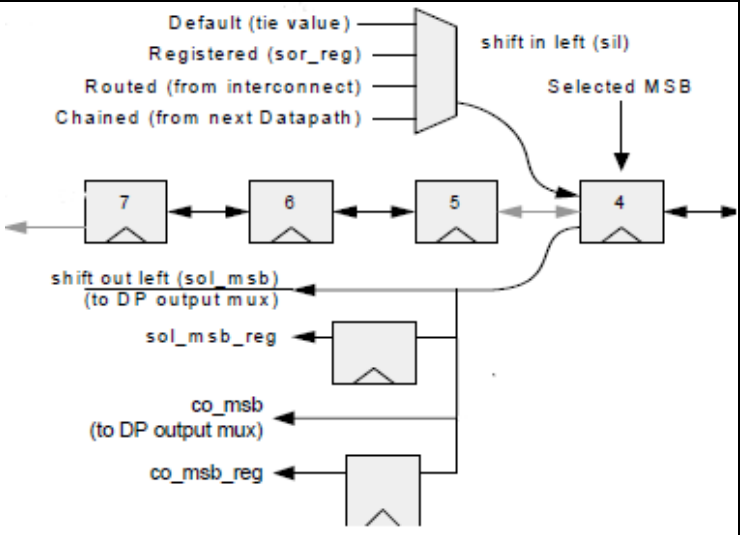
CFG15 and CFG14 Registers

Table 6. CFG15 and CFG14 Register Definitons

CFG15-14 Configuration Selections	
Datapath Configuration Tool	
CFG15-14	

Reset	Binary Value	PI SEL	SHIFT SEL	PI DYN	MSB SI	F1 INSEL	F0 INSEL	MSB EN	MSB SEL	CHAIN CMSB	CHAIN FB	CHAIN 1	CHAIN 0
	00000000 | 00000000	ACC	SL			BUS	BUS	DSBL	BIT0	NOCHN	NOCHN	NOCHN	NOCHN

CFG15-14 Details	
PI SEL Controls the Mux Input for SRCA. The input can either be sourced by the SRCA option in Dynamic Config or from the parallel input.	**ACC:** Source A is sourced by the selection in the dynamic configuration area. **PIN:** Source A is sourced by the parallel input to the datapath. 
Shift SEL Controls the selection of the shift out signal at the datapath output mux.	**SL:** *sol_msb* chosen for output **SR:** *sor* chosen for output 
PI DYN Enables dynamic control of parallel in (PI) mux selection to the ALU ASRC input.	**DS:** PI mux is controlled statically using the PI SEL bit in this register. **EN:** The PI mux is controlled dynamically (assuming PI SEL is '0'), using the CFB_EN bit in the dynamic RAM. When this bit is set and CFB_EN is a '0', the ALU ASRC input is A0 or A1, when CFB_EN is a '1', the ALU ASRC input is PI routing.
MSB SI Supports arithmetic shift right operation.	**REG:** Default shift in selection is defined by the DEF SI value (CFG12, Table 5). **MSB:** Overrides the default shift in value (defined as '00' selection in SELA[1:0]/SELB[1:0]) with the currently defined MSB (MSB EN and MSB SEL fields in CFG14).

CFG15-14 Details (contd.)	
F1 INSEL & F0 INSEL Defines the input source of FIFO 1 and FIFO 0.	**Bus:** FIFO input is the CPU bus, FIFO output is A0 or D0 Registers **A0:** FIFO input is A0. FIFO Output is CPU BUS. **A1:** FIFO input is A1. FIFO Output is CPU BUS. **ALU:** FIFO input is the ALU. FIFO Output is CPU BUS. 
MSB EN and MSB SEL You can adjust the MSbit of the ALU output. These two settings allow you to disable and enable this feature and choose which bit is the MSbit.	When the MSbit is changed to anything other than bit 7, the shift in, shift out, and carry out outputs all change accordingly. 
Chain CMSB Enables chaining from the next Datapath in the chain to this block for the CRC MSB.	The CRC MSB signal flow is from MS block to LS block. Set this bit when this Datapath does not contain the most significant bit of a CRC computation. **NOCHN:** CRC MSB is not chained. **CHNED:** Chain the CRC MSB from the next Datapath block in the chain.
Chain FB Enables chaining from the previous Datapath in the chain to this block for the CRC feedback.	The CRC FB signal flow is from LS block to MS block. Set this bit when this Datapath does not contain the least significant bit of a CRC computation. **NOCHN:** CRC feedback is not chained. **CHNED:** CRC feedback is chained from the previous Datapath block in the chain.
Chain 1 & Chain 0 Defines whether the outputs of CL0, CL1, CE0, CE1, Z0, Z1, FF0, and FF1 are chained.	When set to chained (CHNED), the conditions from the previous datapath are chained to this datapath. Chain 0 affects CL0, CE0, Z0, and FF0 conditions. Chain 1 affects CL1, CE1, Z1, and FF1 conditions.

CFG17 and CFG16 Registers

Table 7. CFG17 and CFG16 Register Definitions

FIFO Configurations
Datapath Configuration Tool

CFG17-16

Reset	Binary Value	Unused [15:13]	ADD SYNC	Unused [11:10]	F1 DYN	F0 DYN	F1 CK INV	F0 CK INV	FIFO FAST	FIFO CAP	FIFO EDGE	FIFO ASYNC	EXT CRCPRS	WRK16 CONCAT
	00010000 | 00000000	000	ADD	00					DP	AX	LEVEL	SYNC	DSBL	DSBL

- ADD SYNC – Determines whether an additional sync flip-flop is added to the FIFO block status. This controls the cycle timing between bus reads/writes at bus clock resolution, and the assertion of the new status on Datapath output routing. There is only one configuration bit that controls this for both FIFOs. See the FIFO Configurations section on the next page.

  NONE: Does not add a flip-flop to the output of the FIFO block status.
  ADD: Adds a flip-flop to the output of the FIFO block status.

- F1 DYN – Controls whether the FIFO1 direction is static or dynamic. In static mode, the F1_SEL[1:0] bits control the FIFO read and write access. When this bit is set for dynamic mode there are two configurations: internal access, where the FIFO can be read and written to by the Datapath, and external access, where the FIFO can be read and written to by the system bus. In this mode, the F1_SEL[1:0] bits control the FIFO write source in internal access mode.

  OFF: Static Mode. FIFO direction is static and controlled by F1_SEL[1:0].
  ON: Dynamic Mode. FIFO direction is dynamic and controlled between internal and external access by toggling the DP routed signal d1_load.

- F0 DYN – Read description for F1 DYN

- F0 CLK INV and F1 CLK INV – Determine whether the FIFO clock is inverted relative to the datapath clock.

  NEG: FIFO clock is the same polarity as the DP clock
  POS: FIFO clock is inverted with respect to the DP clock

- FIFO FAST – Determines whether the FIFOs are clocked using the Datapath clock or the PSoC Bus Clock. In fast mode, the FIFO is clocked by the bus clock, which reduces capture latency.
  The use of this mode results in slightly higher power consumption because the master and quadrant gating of bus clock must be enabled. This bit controls the mode for both FIFOs in the UDB, but it only applies to FIFOs that are configured in output mode.

  DP: Datapath Clock
  BUS: Bus Clock

- FIFO CAP – Enables FIFO capture mode. If enabled, a read of A0 or A1 will write into F0 or F1, respectively.

  AX: A read of A0 or A1 returns the value in the register directly.
  FX: A read of A0 (or A1) triggers a capture into F0 (or F1).

- FIFO EDGE – Determines whether FIFO writes occur on a LOW to HIGH transition. Or, if they can occur at any time, the F0 or F1 load signal is HIGH.

  LEVEL: A FIFO write (output mode) is level sensitive.
  EDGE: A FIFO write (output mode) is edge sensitive.

- FIFO ASYNC – Determines if a flip-flop is needed on the output of the FIFO block status signals. See the FIFO Configurations section on the next page.

  SYNC: Does not add a flip-flop at the output of the FIFO block status.
  ASYNC: Adds a flip-flop to the output of the FIFO block status. Setting ADD SYNC to *NONE* and FIFO ASYNC to *ASYNC* or ADD SYNC to *ADD* and FIFO ASYNC to *SYNC* is acceptable only if the Datapath clock has been synchronized to MASTER_CLK.

- EXT CRCPRS – Overrides the internal configuration for CRC/PRS calculation and allows external routing of CRC/PRS signals. When this bit is set, access is given to the raw block inputs for the CRC operation, including the shift in data and the feedback data, and calculations for these signals must be done externally. (Typically in the PLD).

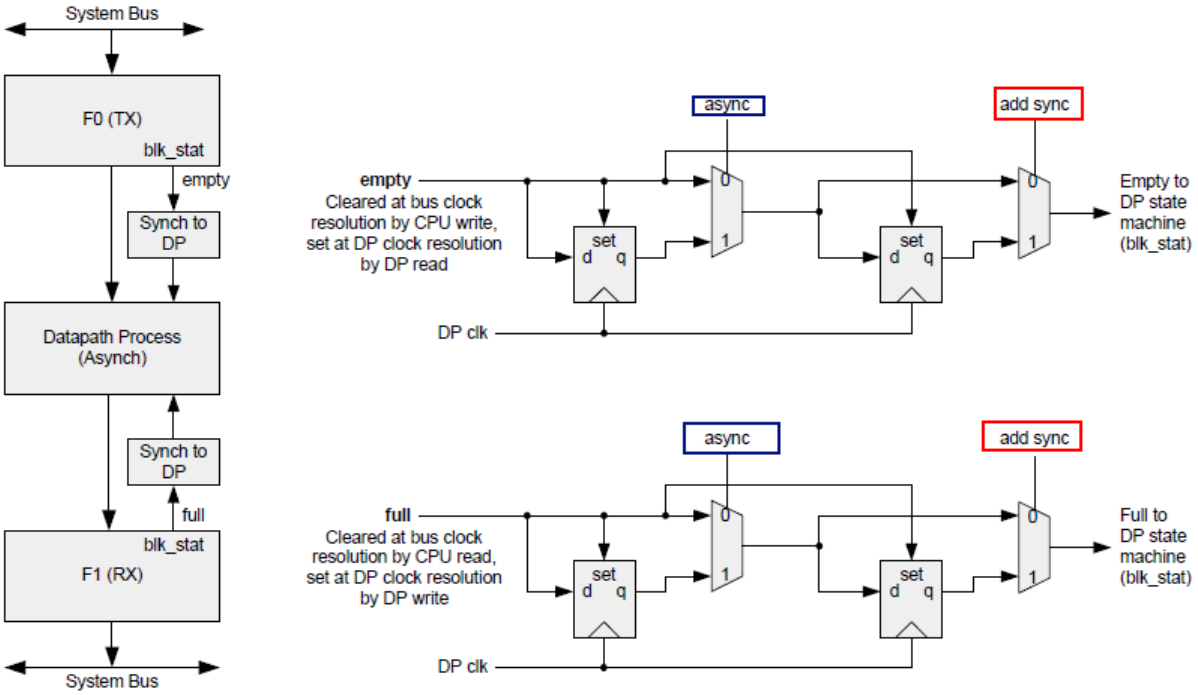
  DSBL: Internal CRC/PRS routing
  ENBL: External CRC/PRS routing.

- WRK16 CONCAT – Controls the working register access mode in the 16-bit access space. By default, when this bit is a '0' the access occurs the same register across a pair of UDBs in chaining order. When this bit is set to '1', a 16-bit read or write accesses concatenated registers within a single UDB. The combinations are {A1,A0}, {D1,D0}, F1,F0}, {CTL,STAT},{MSK, ACTL}, {8'b0,MC}.

  DSBL: 16-bit Default Access Mode. A 16-bit access reads/writes a given register in two consecutive UDBs in chain/address order.
  ENBL: 16-bit Concat Access Mode. A 16-bit access reads/writes concatenated registers in a single UDB.

FIFO Configurations

FIFO Output Synchronization Diagram

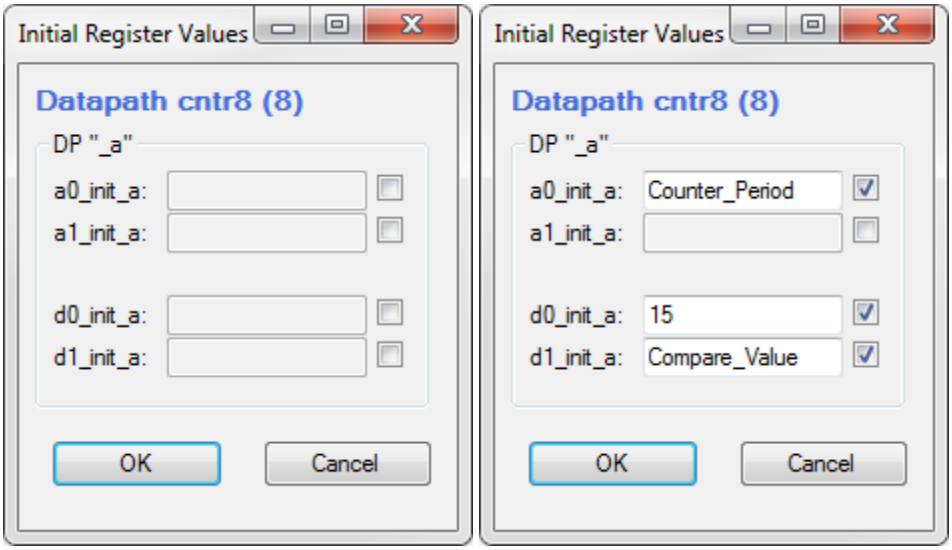


ASYNC	ADD SYNC	Operation	Usage Model
0	0	Synchronous to Bus clock	CPU read/write status changes occur at bus clock resolution. Can be used for minimum latency if the bus clock timing can be met.
0	1	Re-sampled from Bus Clock to DP Clock	This should be the default synchronous operating mode. When the CPU read/write status changes are synchronously re-sampled with the currently selected DP clock. Gives a full cycle of DP clock setup time to the UDB logic.
1	0	(redundant)	
1	1	Double Synced from Bus Clock to DP Clock	When a free running asynchronous DP clock is in use, this setting can be used to double sync the CPU read and write actions to the DP clock.

Setting Initial Register Values

To set the initial values of A0, A1, D0, D1 in the DCT, go to View>Initial Register Values. Say the Datapath name is Cntr8, the window would look like Figure 3 (left).

Figure 3. Initial Register Value



You can set the initial values by clicking on the checkboxes and entering either a number or a valid parameter name from the destination Verilog file (Figure 3 right).

Datapath Chaining

Dedicated Datapath chaining signals allow efficient implementation of single-cycle 16-, 24-, and 32-bit bit functions without the use of channel routing resources.

As shown in Figure 4, all generated conditional and capture signals chain in the direction of least significant to the most significant blocks. Shift left also chains from least to most significant. Shift right chains from most significant to least significant. The CRC/PRS chaining signals of CFBO (feedback) chain least to most, but the CMSBO (MSB output) chains from most to least significant.

Figure 4. Datapath Chaining Signal Flow

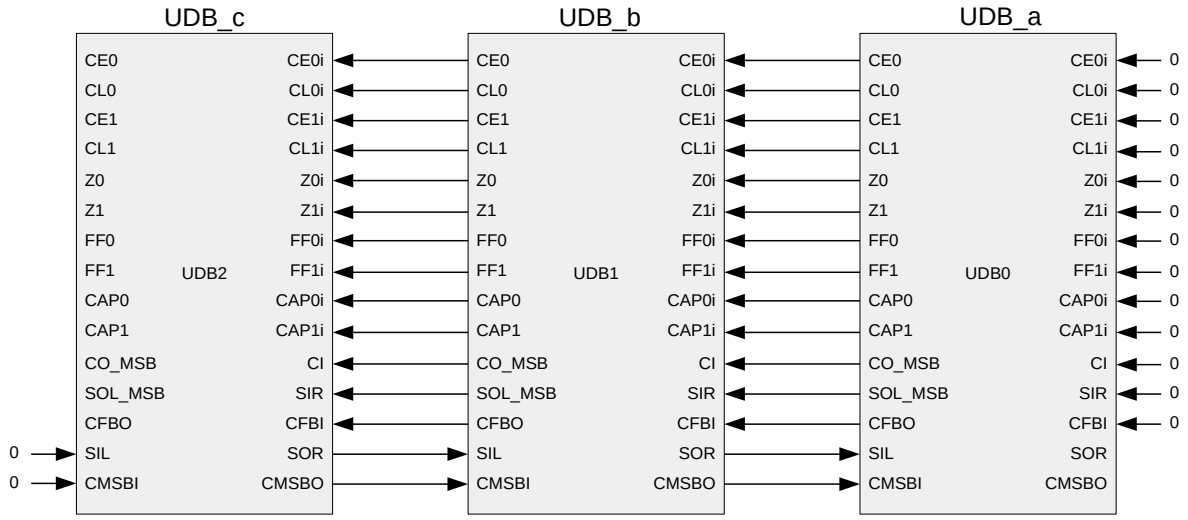
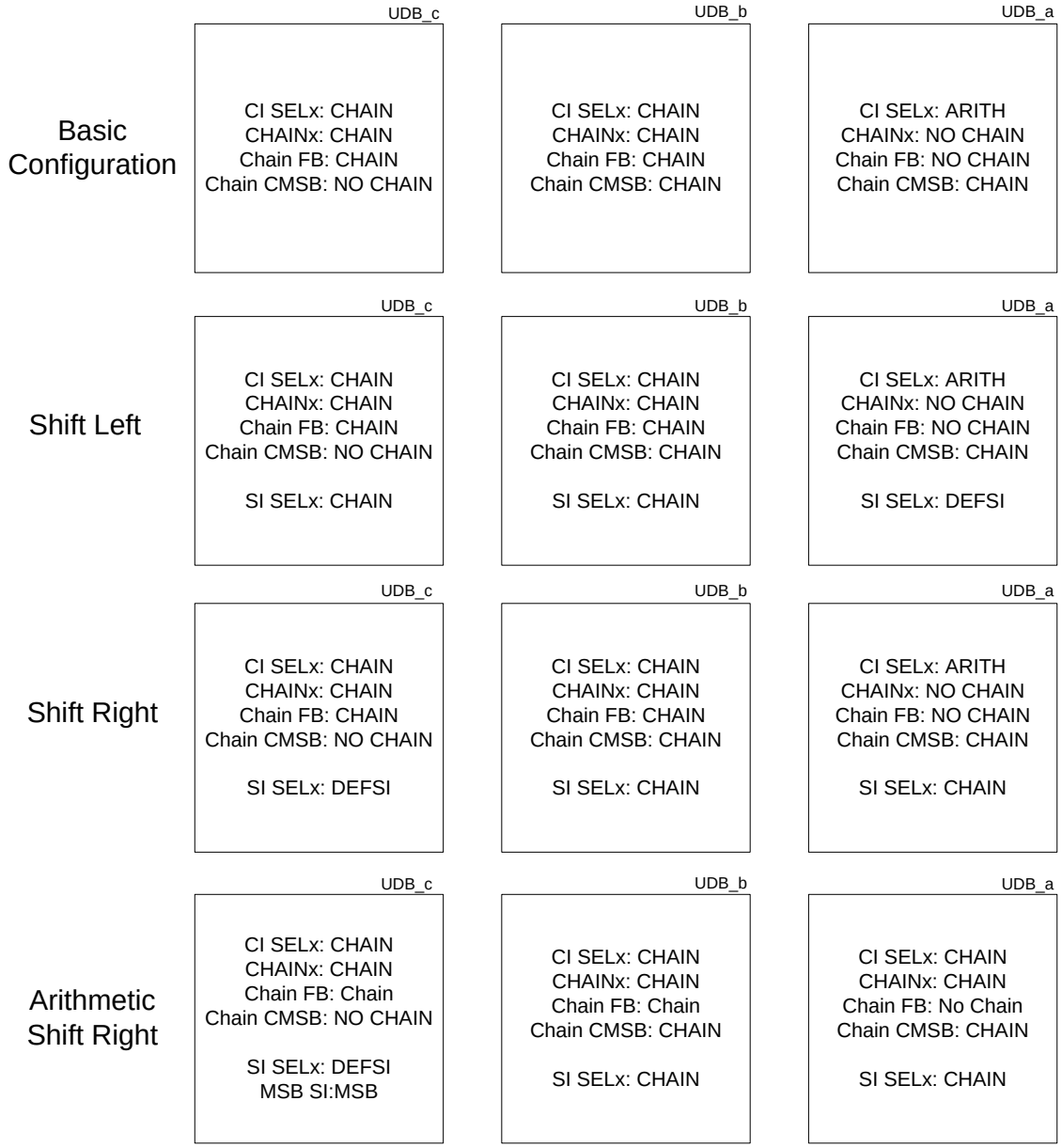


Figure 5 shows the settings required for chaining Datapaths for various cases. UDB_a is the least significant block, while UDB_c is the most significant block. Figure 5 describes a 3 UDB (up to 24-bit) function; a 16-bit or 32-bit function can be created by removing or duplicating the middle Datapath configuration. The figure shows configuration for Chain FB and Chain CMSB even though they may not be used.

Keeping Figure 4 in mind, when chaining together Datapaths, a majority of designs (for example, simple adding or subtracting) would use the 'Basic Configuration' row in Figure 5, that is, chain all signals from LSB (UDB_a) to MSB (UDB_c) except for Chain CMSB. If you perform any shift operations, based on the direction of shift, you need to change the shift chaining configuration – shown in the Shift Left, Shift Right and Arithmetic Shift Right rows in Figure 5.

Figure 5. DCT Configuration for Chaining



Firmware-Control of Datapath Registers

The Datapath registers can be accessed in firmware by using the macros CY_SET_REG8(addr, value) and CY_GET_REG8(addr), or the corresponding 16-, 24-, or 32-bit versions of these functions as the case may be.

The address of the registers can be found in the cyfitter.h file (generated after a successful build). For example, if the 8-bit Datapath instance named cntr8 is instantiated in a component named SimpleCntr8_1, the cyfitter.h file contains a block of code which lists the addresses of all the Datapath registers Figure 6.

```
/* SimpleCntr8_1 */
#define SimpleCntr8_1_cntr8_u0__16BIT_A0_REG CYREG_B1_UDB05_06_A0
#define SimpleCntr8_1_cntr8_u0__16BIT_A1_REG CYREG_B1_UDB05_06_A1
#define SimpleCntr8_1_cntr8_u0__16BIT_D0_REG CYREG_B1_UDB05_06_D0
#define SimpleCntr8_1_cntr8_u0__16BIT_D1_REG CYREG_B1_UDB05_06_D1
#define SimpleCntr8_1_cntr8_u0__16BIT_DP_AUX_CTL_REG CYREG_B1_UDB05_06_ACTL
#define SimpleCntr8_1_cntr8_u0__16BIT_F0_REG CYREG_B1_UDB05_06_F0
#define SimpleCntr8_1_cntr8_u0__16BIT_F1_REG CYREG_B1_UDB05_06_F1
#define SimpleCntr8_1_cntr8_u0__A0_A1_REG CYREG_B1_UDB05_A0_A1
#define SimpleCntr8_1_cntr8_u0__A0_REG CYREG_B1_UDB05_A0
#define SimpleCntr8_1_cntr8_u0__A1_REG CYREG_B1_UDB05_A1
#define SimpleCntr8_1_cntr8_u0__D0_D1_REG CYREG_B1_UDB05_D0_D1
#define SimpleCntr8_1_cntr8_u0__D0_REG CYREG_B1_UDB05_D0
#define SimpleCntr8_1_cntr8_u0__D1_REG CYREG_B1_UDB05_D1
#define SimpleCntr8_1_cntr8_u0__DP_AUX_CTL_REG CYREG_B1_UDB05_ACTL
#define SimpleCntr8_1_cntr8_u0__F0_F1_REG CYREG_B1_UDB05_F0_F1
#define SimpleCntr8_1_cntr8_u0__F0_REG CYREG_B1_UDB05_F0
#define SimpleCntr8_1_cntr8_u0__F1_REG CYREG_B1_UDB05_F1
```

Figure 6. Addresses of Datapath Registers

Miscellaneous

For more information about the Datapath Configuration Tool, see Appendix B of the Component Author Guide, available in the DCT under **Help**>**Documentation**, or in PSoC Creator under **Help**>**Documentation**>**Component Author Guide**.