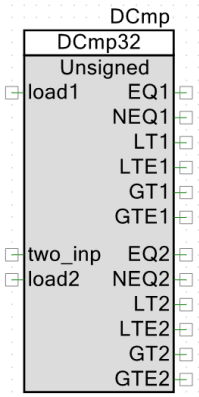
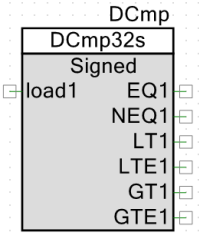
DCmp
1.1.0





Features

- Similar to the Infineon Digital Comparator component but uses significantly fewer PSoC resources.
- HW data-to-threshold comparisons for very fast HW signaling to latter HW stages in your design.
- For use with DMA (preferred) data loading or CPU loading. For example, this can be useful for ADC output count threshold detection.
- 8- to 32-bit data widths for inputs and thresholds.
- Signed comparisons can be selected.
- Design-selectable 6 comparison types EQ (‘=’), NEQ(‘!=’), LT(‘<’),LTE(’<=’),GT(’>’),GTE(’>=’)
- All 6 comparison types **simultaneously** available.
- Optional **simultaneous** dual-threshold comparison values.
- Optional two-input comparison.
- Registered comparison outputs for ‘glitchless’ operation.

General description

The DCmp component implements a very fast HW comparison of input data to threshold data in 8- to 32-bit data widths. One to six of the comparison types are optionally selectable and simultaneously available and synchronized.

When to Use a DCmp

The DCmp component can be used in place of the Infineon Digital Comparator in many instances.

Here is a side-by-side comparison table for each of the two components. (I’ve highlighted what I believe to be the better features):

Table 1. Component Features Compare

Method	Infineon Digital Comparator	CONSUL*TRON* DCmp
Input load	HW	DMA or CPU

Threshold load	HW	DMA or CPU
Number of simultaneous inputs	1	1 or 2
Number of simultaneous comparisons	1	1 or 2
Simultaneous comparison outputs	1 HW	1 to 6 HW types
Data Widths	1- to 32- bits	1- to 32- bits
Comparison output glitch filter	No (async)	Yes (load signal sync)
Relative PSoC resource allocation*	>=(1/3)x@8-bits >=(2/9)x@16-bits >=(2.5/13)x@24-bits >=(3/17)x@32-bits	(1/1)x @ 1- to 8-bits + 1 Datapath (1/1)x @ 9- to 16-bits + 2 Datapath (1/1)x @ 17- to 24-bits + 3 Datapath (1/1)x @ 25- to 32-bits + 4 Datapath
Relative Comparison Speed	Fastest (async)	Very fast using DMA (>4M cmps/sec)
Signed comparisons	No	Yes
* "(mm/uu)x" represents macrocells/unique pterms.		

The main benefits of using the Infineon Digital Comparator component are the comparison speed and the HW data input and threshold inputs. This provides the fastest comparison even compared to the DCmp component. However, if your design can tolerate some minor latency delays, the DCmp component minimizes PSoC resources used, provides more simultaneous comparison types up to two thresholds, two inputs or signed comparisons.

Here's an extreme example of PSoC resource savings of the DCmp component over the Infineon Digital Comparator component:

Infineon Digital Comparator	DCmp
• LT comparison • 32-bits • One data input • One threshold	• EQ, NEQ, LT, LTE, GT, GTE comparisons for both Output sets 1 and 2. • 32-bits • Two data inputs • Two thresholds
MacroCells = 25 Unique PTerms = 151 Datapath Cells = 0	MacroCells = 16 Unique PTerms = 15 Datapath Cells = 4

Device Families Supported

At this time, the following are the PSoC devices that support this component:

- PSoC5

Implementation Limitations

The DCmp component is only capable of up to 32-bit comparisons.

Choosing signed comparisons will disallow dual-threshold and two input comparisons.

The DCmp component is more effective if data loading is done by DMA. CPU data loading is achievable but the comparison latency is much longer and a CPU math comparison might be more effective.

Input-Output connections

DCmp Component has selectable inputs to control the loading of data into the component.

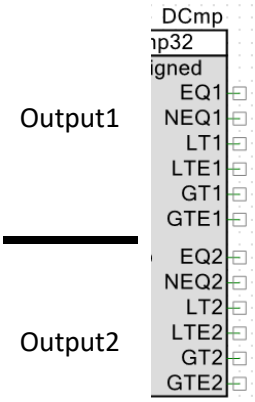
Inputs

load1 – This input is always present. This input is the signal from your design that Data1 was loaded into the component via DMA or CPU and the rising edge of this signal performs the comparison and latches it to the outputs.

two_inp – This input is available if the "Signed Compare" is not selected and "Two input DCmps" and "DCmp2 outputs enable" are selected. This input signals to the HW that Threshold2 will be used on loaded Data2 and the comparison results will be present on any of the selected types of Output2. The input load2 will be also available.

load2 – This input is present when "Two input DCmps" is selected. This input is the signal from your design that Data2 was loaded into the component via DMA or CPU and the rising edge of this signal performs the comparison and latches it to Output2 types.

Outputs



The second set of comparison outputs (Output2) will only be present if the component is using unsigned comparisons and the "DCmp2 outputs enable" is selected. If an enable for a comparison listed below is not selected, that output connection will not be present.

EQ1 and EQ2 – The EQ output is available if the "Enable EQ output x" is selected for Output1 and/or Output2.

NEQ1 and NEQ2 – The NEQ output is available if the "Enable NEQ output x" is selected for Output1 and/or Output2.

LT1 and LT2 – The LT output is available if the "Enable LT output x" is selected for Output1 and/or Output2.

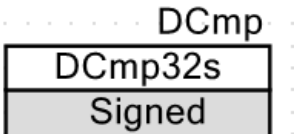
LTE1 and LTE2 – The LTE output is available if the "Enable LTE output x" is selected for Output1 and/or Output2.

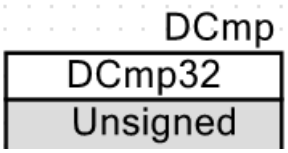
GT1 and GT2 – The GT output is available if the "Enable GT output x" is selected for Output1 and/or Output2.

GTE1 and GTE2 – The GTE output is available if the "Enable GTE output x" is selected for Output1 and/or Output2.

Parameters and Settings

Basic tab





Data Width – This parameter sets the data width of the comparison.

Signed Compare – This parameter allows for signed value comparisons. If selected, only one input, one threshold (Thresh1) and one set of comparison outputs are used (Output1).

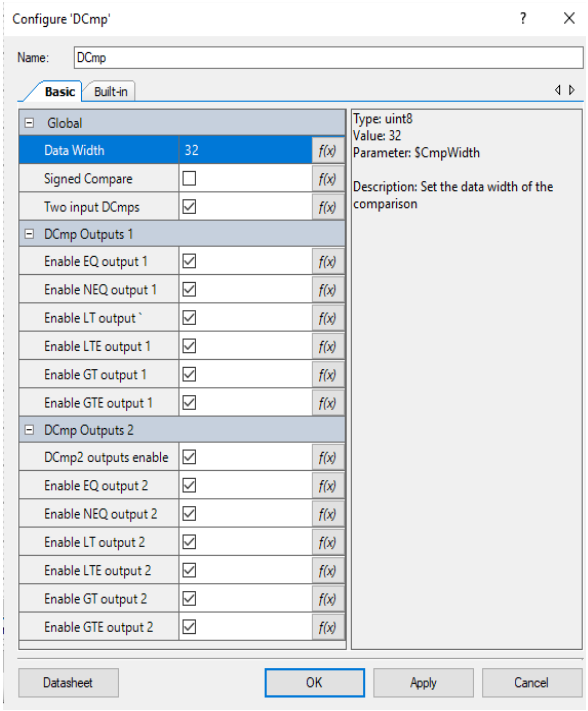
Two input DCmps – This parameter allows for two separate inputs to be compared.

If not selected, only one input is used (Data1). Data1 uses Thresh1 for the comparisons and are sent to Output1 outputs and Thresh2 for the comparisons and are sent to Output2 outputs.

If selected, two inputs are used. Data1 uses Thresh1 for the comparisons and are sent to Output1 outputs. Data2 uses Thresh2 for the comparisons and are sent to Output2 outputs.

This selection is not available if “Signed Compare” is selected or “DCmp2 outputs enable” is unselected.

DCmp Outputs 1 =>



Enable <cmptype> output 1 – This parameter selects which of the Output1 comparison types to output. If not selected, the output terminal is not present.

DCmp Outputs 2 =>

DCmp2 outputs enable – This parameter selects if any of the Output2 set of comparison outputs are available. If not selected, all the output terminals associated with this output group are not present.

This selection is not available if “Signed Compare” is selected.

Enable <cmptype> output2 – This parameter selects which of the Output2 comparison types to output. If not selected, the output terminal is not present.

These selections are not available if “Signed Compare” is selected or “DCmp2 outputs enable” is unselected.

DCmp Special Considerations

Clock Requirements

The DCmp component is comprised of UDB datapaths and logic which is limited in comparison speed by the BUS_CLK frequency. BUS_CLK is used internally to clock internal state logic.

The comparison outputs are latches by the rising edge of the load1 (or load2) inputs **after** the inputs (or thresholds) are loaded into the component registers. This prevents combinatorial comparison 'glitches' on the output(s).

If faster comparison speed is needed, you may be able to increase the BUS_CLK up to the PSoC maximum allowed. However this might cause the PSoC Creator might complain about setup or hold time violations. Allow these violation warnings do not prevent a build from succeeding, I recommend using a "Sync" component (or D flip-flop) in the load signal path to address the warning. Doing so will also add one BUS_CLK period as latency.

Application Programming Interface (API)

The following table is a list of the supported API calls supported.

Table 1 - Supported API calls

Function	Description
DCmp_Init()	Initializes the component with certain customizer values
DCmp_StartEx()	Starts the component with extended values
DCmp_Stop()	Stops the component. (No effect in this version).
DCmp_Enable()	Enables the component. (No effect in this version).
DCmp_SetData1()	Sets the Data1 value
DCmp_GetData1()	Gets the Data1 value
DCmp_SetThresh1()	Sets the Threshold1 value
DCmp_GetThresh1()	Gets the Threshold1 value
DCmp_SetData2()	Sets the Data2 value
DCmp_GetData2()	Gets the Data2 value
DCmp_SetThresh2()	Sets the Threshold2 value
DCmp_GetThresh2()	Gets the Threshold2 value
DCmp_GetNumInputs()	Gets the number of inputs in use.
DCmp_GetCmpOut1Types()	Gets the comparison types of Output1 selected.
DCmp_GetCmpOut2Types()	Gets the comparison types of Output2 selected.
DCmp_GetCmpParams()	Gets the comparison operational parameters

Global Variables and defines

Variable	Description
DCmp_initVar	Indicates whether the DCmp has been initialized. The variable is initialized to 0 and set to 1 the first time DCmp_Start() is called. This allows the component to restart without reinitialization after the first call to the DCmp_Start() routine. If reinitialization of the component is required, then the DCmp_Init() function can be called before the DCmp_Start() or DCmp_Enable() function.

DCmp_DPWIDTH	Define of the size of the data path.
DCmp_CMPWIDTH	Define of the size of the data width of the comparisons.
DCmp_SIGNED	Define of whether signed comparisons are used.
DCmp_DP_SZ_TYPE	Define of the Data1, Data2, Thresh1, and Thresh2 size types based on DCmp_DPWIDTH. This define is sized based on the UDB datapath allocated. 1- to 8-bits => uint8 or int8, 9- to 16-bits => uint16 or int16, 17- to 24-bit => unit32 to int32 and 25- to 32-bit => uint32 or int32.

void DCmp_Init(void)

Description: Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call DCmp_Init() because the DCmp_Start() API calls this function and is the preferred method to begin component operation.
Parameters: None
Return Value: None

void DCmp_Start(void)

Description: Performs all of the required initialization for the component and enables power to the block. The first time the routine is executed, the component is initialized to the configured settings. When called to restart the DCMP following a DCMP_Stop() call, the current component parameter settings are retained.
Parameters: None
Return Value: None
Side Effects: If the DCMP_initVar variable is already set, this function only calls the DCMP_Enable() function.

void DCmp_StartEx(DCmp_DP_SZ_TYPE Thresh1, DCmp_DP_SZ_TYPE Thresh2)

Description: Operates like DCmp_Start() and calls DCmp_Init() and loads new values into Thresh1 and Thresh2.
Parameters: DCmp_DP_SZ_TYPE Thresh1
DCmp_DP_SZ_TYPE Thresh2
Return Value: None
Side Effects: If the DCMP_initVar variable is already set, this function only calls the DCMP_Enable() function.

void DCmp_Stop(void)

Description: Stops the component. Not used in this design.
Parameters: None
Return Value: None

void DCmp_Enable(void)

Description: Activates the hardware and begins component operation. Not used in this design.
Parameters: None
Return Value: None

void DCmp_SetData1(DCmp_DP_SZ_TYPE value)

void DCmp_SetData2(DCmp_DP_SZ_TYPE value)

Description: Sets the DCmp Data1 or Data2 with the value supplied.
This is only useful for CPU loading.
This function for Data2 will do nothing if DCmp_SIGNED == 1 or DCmp_TWO_INP_EN == 0.
Parameters: (DCmp_DP_SZ_TYPE) value
Return Value: **None**

DCmp_DP_SZ_TYPE DCmp_GetData1(void)
DCmp_DP_SZ_TYPE DCmp_GetData2(void)

Description: Sets the DCmp Data1 or Data2 with the value supplied.
This is only useful for CPU evaluation.
This function for Data2 will return 0 if DCmp_SIGNED == 1 or DCmp_TWO_INP_EN == 0.
Parameters: void
Return Value: (DCmp_DP_SZ_TYPE) value

void DCmp_SetThresh1(DCmp_DP_SZ_TYPE value)
void DCmp_SetThresh2(DCmp_DP_SZ_TYPE value)

Description: Sets the DCmp Thresh1 or Thresh2 with the value supplied.
This is only useful for CPU loading of the desired threshold values for the comparisons.
This function for Thresh2 will do nothing if DCmp_SIGNED == 1.
Parameters: (DCmp_DP_SZ_TYPE) value
Return Value: **None**

DCmp_DP_SZ_TYPE DCmp_GetThresh1(void)
DCmp_DP_SZ_TYPE DCmp_GetThresh2(void)

Description: Sets the DCmp Thresh1 or Thresh2 with the value supplied.
This is only useful for CPU evaluation.
This function for Thresh2 will return 0 if DCmp_SIGNED == 1.
Parameters: void
Return Value: (DCmp_DP_SZ_TYPE) value

uint8 DCmp_GetNumInputs(void)

Description: Gets the number of inputs available.
Parameters: void
Return Value: (uint8) number of inputs available

uint8 DCmp_GetCmpOut1Types (void)

Description: Gets the current comparison Output1 types being defined. Each bit refers to the types selected.
EQ => bit 0; NEQ => bit 1; LT => bit 2; LTE => bit 3; GT => bit 4; GTE => bit 5
Parameters: None
Return Value: (uint8) The comparison types selected

uint8 DCmp_GetCmpOut2Types (void)

Description: Gets the current comparison Output2 types being defined. Each bit refers to the types

selected.
EQ => bit 0; NEQ => bit 1; LT => bit 2; LTE => bit 3; GT => bit 4; GTE => bit 5
This function for will return 0 if DCmp_SIGNED == 1 or or DCmp_TWO_INP_EN == 0.

Parameters: None

Return Value: (uint8) The comparison types selected

uint32 DCmp_GetCmpParams (void)

Description: Gets the current operational parameters.
Byte[0]:
Output2 Enabled => bit 0
Two Inputs selected => bit 1
Signed comparisons => bit 2
Byte[1]:
Number of bits of the comparison.
Byte[2]:
Number of bits of the datapath allocated.

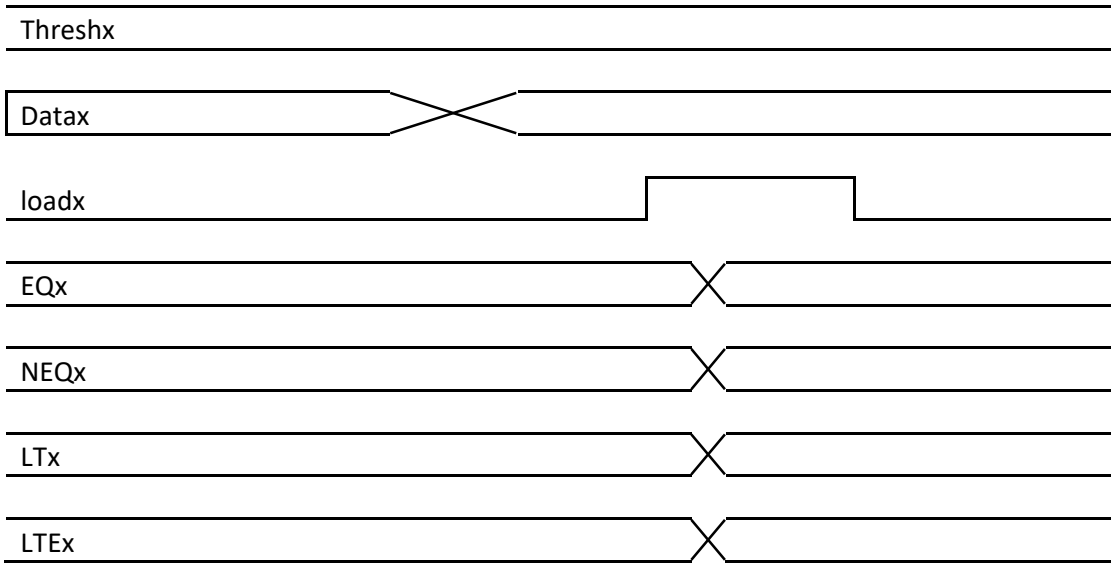
Parameters: None

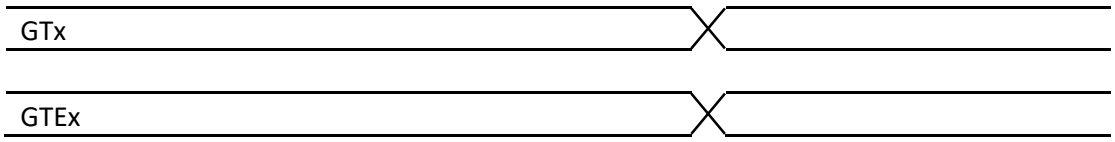
Return Value: (uint32) The operational parameters.

Functional Description

The implementation of the DCmp component was intended to make improvements to the Infineon Digital Comparator component. The improvements are to significantly minimize PSoC resources. It also allows for signed comparisons or simultaneous dual-threshold or two input comparisons with multiple comparison types.

The data to be compared needs to be loaded into the component Data1 or Data2 (for two input mode) first using the DMA (preferred) or CPU. Then the load1 (or load2 for Data2) rising edge informs the HW to compare the input data to the Threshold values loaded earlier (Thresh1 and Thresh2) and latch the comparison results into the output for 'glitch-free' HW signals to be used in downstream circuits.





If using the DMA to load Data1 or Data2, the "NRQ" output can be used to signal to DCmp that the Data has completed loading and to perform the comparison.

Relative Performance

The Infineon Digital Comparator component can resolve a comparison as fast as the macrocell and P-Term logic can execute which within about two propagation delays (~ 20ns). However the comparison is combinatorial which can be prone to glitches. Placing a clocked D-Flip-flop after the output should eliminate the glitches but adds latency.

The DCmp component performance using DMA is about 1/12th of the BUS_CLK. This is because the latency is dependent on the DMA grabbing the CPU bus and can take 8 to 10 BUS_CLK cycles to move the data. Then there are a couple of BUS_CLK cycles to sync all the outputs without glitches.

The DCmp component performance using CPU is about 1/10th of the DMA performance. It is highly dependent on the BUS_CLK frequency and other interrupt or DMA processing which will increase the latency.

Resources

The DCmp component utilizes the following resources.

	Resource Type						
Data Width (bits)	***Datapath Cells***	***Macro Cells***	***Unique PTerms***	***Status Cells***	***Control Cells***	***FLASH (bytes)***	***RAM (bytes)***
1 to 8	*1*	*<=16*	*<=15*	*0*	*0*	*<=240*	*1*
9 to 16	*2*	*<=16*	*<=15*	*0*	*0*	*<=240*	*1*
17 to 24	*3*	*<=16*	*<=15*	*0*	*0*	*<=288*	*1*
25 to 32	*4*	*<=16*	*<=15*	*0*	*0*	*<=310*	*1*

The actual MacroCells and Unique PTerms used are not dependent on data width. They are based on the number of comparison types selected as well as if Output 2 and if two inputs are used.

The Table below displays the number of these resources based on which comparison types are selected. In general, the fewer types selected, the lower the number of MacroCells and Unique PTerms used. The "ALL OUT 1" column indicates the maximum resources for all types selected. The "ALL OUT 2" column indicates the added resources for all types selected.

Table 2- UDB MacroCell and Unique PTerms Used

Signed

Data Width (bits)	EQ1		NEQ1		LT1		LTE1		GT1		GTE1		ALL OUT1	
	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT
1 to 32	5	6	5	6	5	9	5	9	5	9	5	9	10	19

Unsigned two_inp enabled = No

Data Width (bits)	EQ1		NEQ1		LT1		LTE1		GT1		GTE1		ALL OUT1	
	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT
1 to 32	4	4	4	4	4	4	4	4	4	4	4	4	9	8

	EQ2		NEQ2		LT2		LTE2		GT2		GTE2		ALL OUT2	
	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT
1 to 32	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+6	+5

Unsigned two_inp enabled = Yes

Data Width (bits)	EQ1		NEQ1		LT1		LTE1		GT1		GTE1		ALL OUT1	
	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT
1 to 32	4	4	4	4	4	4	4	4	4	4	4	4	9	8

	EQ2		NEQ2		LT2		LTE2		GT2		GTE2		ALL OUT2	
	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT	MC	UPT
1 to 32	+2	+3	+2	+3	+2	+3	+2	+3	+2	+3	+2	+3	+7	+7

MC = MacroCells UPT = Unique PTerms.

Sample Firmware Source Code

I have provided a ***DCmp_s_Demo*** example to exercise many of the API calls and demonstrate the features of DCmp.

Expected Results

The demo program uses menu commands to change some DCmp parameters at run-time.

Component Changes

Version	Description of changes	Reason for changes/impact
1.0.0	first release of the component	

References

DCmp_s_Demo_1_0_0.pdf	