CYPRESS SEMICONDUCTOR CORPORATION
Internal Correspondence

Date: Thursday 1/10/2013 **WW: 1302**
To: **PSoC Apps**
Author: Chris Keeser (KEES)
Author File#: KEES#184
Subject: **UDB SumCompare Component**
Distribution: PSoC Apps

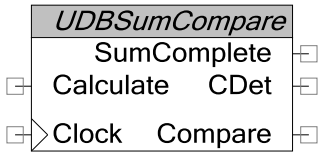
Summary:

This memo documents the UDB based sum and compare component for the FSK Demodulator. This _may_ have use outside of the scope of an FSK demodulator. The purpose of this component is to take 4 values loaded into the the datapath with DMA, add specific pairs together and compare relative magnitudes. The summed pairs are then combined to construct a received signal strength, which is compared to an configurable threshold to assert a carrier detect signal. The compare value and carrier detect are sent to a UART for final conversion from a bitstream into received words. The DMA wizard is also aware of the useful registers, making it easy to configure DMA to move data into and out of the component.

Attached File Summary:
File # 2 is the component archive, created using Creator 2.2's component export feature. To use the file, download it and remove the .PDF extension from the file

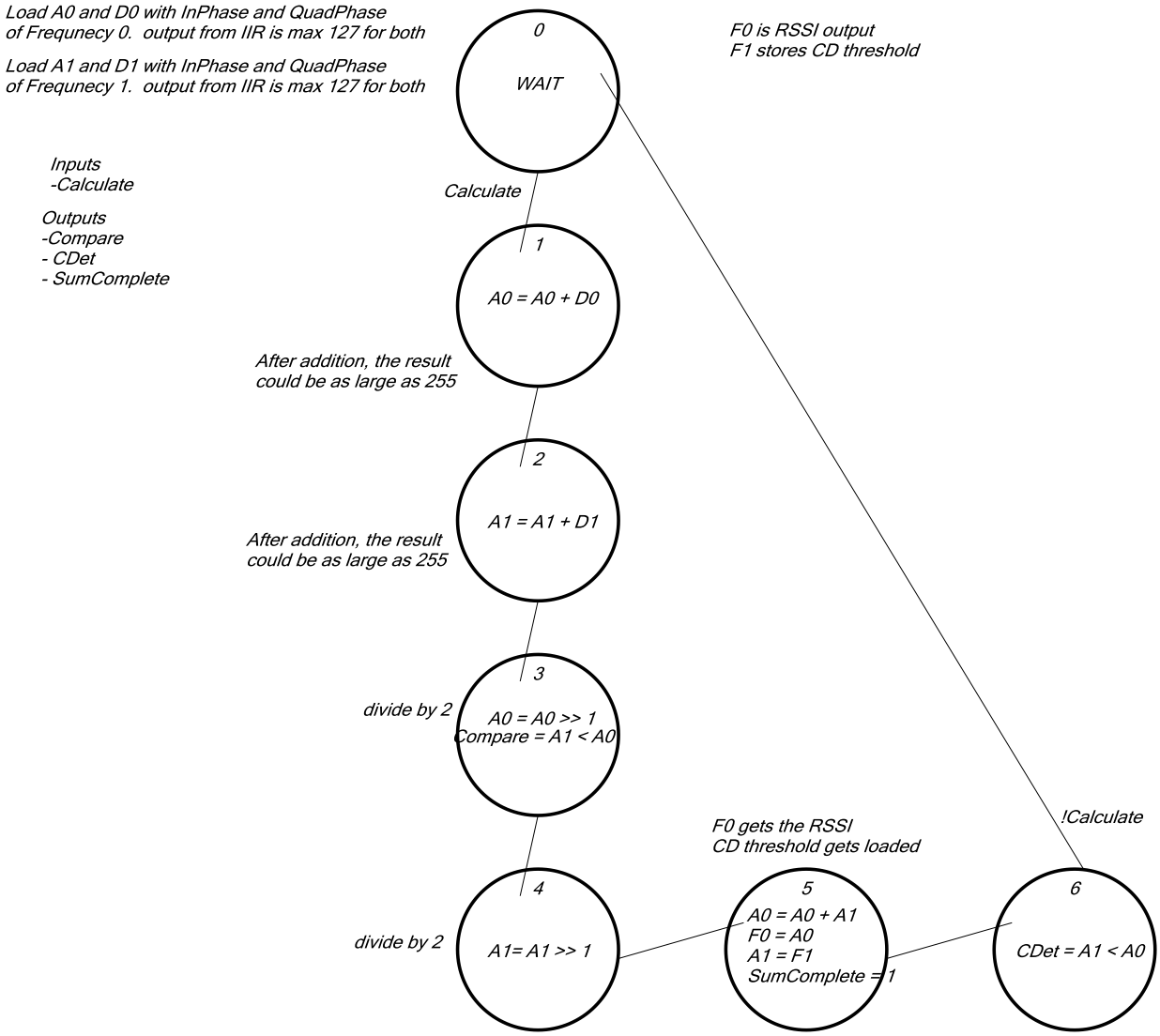
Details:

KEES_UDBSumCompare_1



This datapath adds the absolute value of A0 and D0 together and stored in A0 (the absolute value calculation is performed in the UDB IIR filter automaticall) to approximate the square root of the sum of squares for the in phase and quadrature phase outputs of frequency 0. Each value has a maximum of 127, so overflow is not a concern. The same operation is performed on A1 and D1 and stored in A1 for the energy in frequency 1. These two values are compared

(A0 > A1) to produce a decision of a logic ‘0’ or ‘1’. The signal can be sent to a UART for decoding. A0 and A1 are then shifted right by 1 (to perform a divide by 2) making a maximum value of 127 for each, then the results are summed (A0 + A1) and stored into A0 and FIFO 0 (as a Received Signal Strength Indicator or RSSI) while a carrier detection threshold is loaded from FIFO 1 into A1. The value in A0 is then compared to A1 to see if the received signal strength is greater than the threshold. This signal can be used to gate the UART from decoding anything when the signal threshold is too low. The RSSI signal loaded into F0 can then be read out when the SumComplete signal is asserted, and sent to a VDAC for a real time representation of signal strength, or it can be used in firmware for some other purpose. It is worth noting that I was able to use a FIFO as a constant storage register by pre-filling it with the constant value. When the FIFO underflowed, it continued to pull the constant from the last position in the FIFO. This required the addition of a _Start(); API to pre-fill the FIFO, but the benefit was worth the minor cost.



I used DMA to move the results of the IIR filters into the datapath, then used the NRQ of the DMA channel to trigger the calculation: