Full-custom design of split-set data weighted averaging with output register for jitter suppression

MC Jubay and OJ Gerasta
Mindanao-State University-Iligan Institute of Technology
Iligan City, Philippines

E-mail: mark.jubay@analog.com, olgajoy.labajo@msuiit.edu.ph

Abstract. A full-custom design of an element selection algorithm, named as Split-set Data Weighted Averaging (SDWA) is implemented in 90nm CMOS Technology Synopsys Library. SDWA is applied in seven unit elements (3-bit) using a thermometer-coded input. Split-set DWA is an improved DWA algorithm which caters the requirement for randomization along with long-term equal element usage. Randomization and equal element-usage improve the spectral response of the unit elements due to higher Spurious-free dynamic range (SFDR) and without significantly degrading signal-to-noise ratio (SNR). Since a full-custom, the design is brought to transistor-level and the chip custom layout is also provided, having a total area of 0.3mm², a power consumption of 0.566 mW, and simulated at 50MHz clock frequency. On this implementation, SDWA is successfully derived and improved by introducing a register at the output that suppresses the jitter introduced at the final stage due to switching loops and successive delays.

1. Introduction
As the demand for the increase of performance of digital-to-analog converters (DAC’s) continues to grow, several optimization schemes have been proposed. One factor worth considering is the problem regarding component mismatch, which is basically due to non-idealities introduced in the actual manufacturing process (e.g., inaccuracy and process variation) [1]. Matching effects gained importance as process dimension decreases and power supply lowers [2]. An existing method to cope with this problem is through static means (e.g., application of laser technology and some digital error correction techniques). These schemes have been effective in addressing the problem, but on the other hand can be impractical to someapplications [3]. Having these limitations of static matching, dynamic system became into an option [4]. Dynamic approach can only be applied in equally-weighted elements. This is because the weights are equal and thus they can be interchanged or used independently. This type employs equally-weighted current or voltage units and the number of the unit elements used directly comprises its resolution. Such implementation can potentially result to more circuit complexity, but on the other hand, it is still preferred on some applications due to its advantages (e.g. guaranteed monotonicity, reduced glitching noise and linearity). They are applied in ADC and DAC. The output of an ADC which is in thermometer bits can be fed directly to the input of the DAC unless conversions are necessary. Though this is possible, doing so would consequently

1 To whom any correspondence should be addressed.
degrade the output of the DAC in the presence of element mismatch. Unequal and nonrandomized selection of the unit elements are the drawbacks of using directly a thermometer code. Due to these disadvantages, a basic DWA algorithm is commonly implemented for averaging. Basic DWA algorithm though provides averaging by cyclic usage of the unit elements in time. This, on the other hand, suffers the problem on some type of input since no randomization is done in the selection. Switching the elements equally would be necessary in the reconstruction of the discrete analog signal. This will make the errors spread out over the frequency spectrum resulting to nonlinearities averaged out over time [4]. Other than this, randomization breaks up the input-dependent tones generated by the selection process itself [1,4]. Considering these two criteria, distortion in the frequency domain is reduced in such a way that the SNR is not significantly affected or degraded. SDWA provides the averaging and randomization at the same time by using the DWA output and splitting the elements independently.

In this implementation, Split-set DWA is designed in full-custom. This allows more freedom to work to the lowest design level with the purpose of optimizing the layout area, minimizing glitches or delay problems, and lessening the overall power consumption. Timing limitations that result to glitches and jitter problems are compensated using a register. This register is used to block or filter out the glitches or jitter that occur at the output due to successive digital switching.

2. Experimental section

2.1. SDWA algorithm
SDWA operates by splitting the unit element set into subsets in a special way, and randomizing each subset independently. For N elements, SDWA is carried out in the following steps:

i. Apply DWA to the N-unit elements of DAC for (M-1) clock cycles;

ii. In clock cycle M (where M may be predetermined, or identified by a pseudorandom digital signal reaching a predetermined value), split the set of all unit elements into two subsets. Subset \( S_K \) contains elements 1 through k, where k is the highest unit element index used in clock cycle M: its complement \( S_{K,\text{BAR}} \) contains elements with indices k+1 through N;

iii. Rotate or scramble all elements of \( S_K \) within subset \( S_K \) and similarly rearrange the elements of \( S_{K,\text{BAR}} \) internally within \( S_{K,\text{BAR}} \);

iv. Return to step 1, starting with the unit element now occupying position k+1.

2.2. Design architecture
The implementation of SDWA algorithm is composed of a basic DWA block and some complementing blocks.

![Figure 1. SDWA Circuit Block Diagram.](image-url)
2.2.1. Data weighted averaging

The implementation of the Basic DWA uses logarithmic shifter which is controlled by the output of the accumulator block. Thermometer-to-binary converter provides the encoding of the input data to be able to generate the correct pointer or input value to the logarithmic shifter.

The averaging characteristic of the basic DWA is induced in SDWA by using the DWA as the enabling data of the decoder-OR gate block. The splitting of the unit elements into subsets are provided by the subset shifter block at the control of the pointer decoder data. The updated subset shifter output data is controlled by the shift input which can be a predetermined scrambling-enable data. When this input data is not at enable state, the output of the SDWA block will just be a basic DWA. The unit element register block stores any update in the subset shifter output at every clock period.

3. Experimental section

Fig. 4 above shows the SDWA testbench with 7-bit thermometer code as input. It also requires a clocking signal and a shift enable input. The shift input defines the SDWA scrambling frequency and which can be predefined or identified by a pseudorandom digital signal reaching a predetermined value [3]. For verification, we let the shift enable input to be equal as the clock. This means that SDWA is performed in every clock period. Shift input differentiates DWA to SDWA. When shift is enabled (high), the output is SDWA; when shift is low, output is DWA. All the following simulations are done in 10 pF output load, supply voltage of 1.2 volts having clock frequency at 50MHz.

3.1. SDWA simulation for DC input (constant 2 for 16 cycles)

A constant input sequence of two levels is maintained for 16 clock periods and is inputted to the SDWA circuit is shown in Fig. 6. In fig. 6, the first 2 less significant bits are used for 16 clock periods. This input can be fed directly to the unit elements for reconstruction but later on will provide significant problems in the presence of mismatch. Other than this, it is logical that the first two unit elements will be overused compared to the other bits (bits 3 to 7). This characteristic of thermometer
code that prioritizes the less significant bit in usage can result to potential problems when non-idealities are being introduced during reconstruction.

As a first major process in SDWA, it requires the input to undergo data weighted averaging (DWA). As shown in Fig. 7, DWA distributes the usage of the unit elements through cyclic assignment. Since the input has 2 bits in high for 16 clock periods, the output does the same; only that the assignment is different. Through cyclic assignment of DWA, each bit will be used almost equal as the others. This algorithm makes the long term average usage of the unit elements equal. Output is less susceptible to the effect of mismatch due to errors are averaged as well.

For the purpose of analysis, we provide the DWA block shown in Fig. 2 and Fig. 3. In this way we can picture out and trace the logic level at each different block.

The unit element registers inside the unit element register block are initialized to a particular position value defined by 3-bit data. What determines the position is the value of the 3-bit data itself. Initially, each 3-bit register is set to hold a distinct value from 001 to 111 at the starting time. As explained in the previous chapter, we implement this initialization by introducing delay from the supply to the preset or clear of that specific register. At the first thermometer input, the shift data of the subset shifter is not yet enabled. This does not provide any scrambling or rotation of the initialized position value; thus, the initial position is moved to the output and then is fed to the SDWA output block. Using the obtained DWA data at the first thermo input, un-shifted position data is now is used as the reference position of the SDWA out block at the input. At this period, DWA data is 0000011 and this enables the first two decoders. Since the first and second decoders have obtained its input from the un-shifted data of the unit element register which are 001 and 010 respectively. Using DWA data 0000011 will provide SDWA output of 0000011. It should be noted that at any clock period, each decoder at the input of SDWA out block must have a distinct position value (001 to 111). These position values will not reach the output unless enabled by the DWA data. When shift data of subset shifter is enabled, randomization is done while the SDWA output block maintains the long term equal element usage. We note that in every complete cycle of the thermo input, the subset shifter provides the randomization while the SDWA output block does the long term equal elements usage. For the rest of the clock cycles, we refer to Fig. 11.

At the initial condition, we do not allow any shifting in the logarithmic shifter. This can be realized by initializing the pointer register inside the accumulator at 000 (output). Having no shifting, the thermo input (0000011) is preserved at the output. It is to be noted that clocking will only start at the

![Figure 6. SDWA DC input.](image)
![Figure 7. Generated DWA data for the DC input).](image)

![Figure 8. Accumulator.](image)
![Figure 9. Complementing blocks.](image)
second period of the thermo input; so that the register will not yet store the output from the Modulo-7 Adder. Prior to the first clock cycle, the Modulo-7 adder has just read the output of the thermo-to-binary converter which is 010 (2 in decimal). At the first clock period, the output of the thermo-to-binary converter passes the modulo-7 adder having an addend of (000) and the output is loaded to the pointer register. Since this loaded value is equal to the first thermo input, the logarithmic shifter will provide shifting levels equal to the input. Doing this, the following bit/s will have a distinct assignment in position. In other words, it is not allowed to reuse the same bit position. Any incoming data will be accumulated but the output only rotates from 000-111. Thus any overflow will just be looped back to the input to maintain DWA. The system is just like playing with cards and distributing them in cyclic way to number of people.

![Figure 10. Output Verification (periods 1-16).](image)

Logic level analysis is shown in Figure 11 for 16 periods of thermometer input. Note that P (position), D (DWA), and S (SDWA) are used as identifiers. Looking in the first period, position data are initialized from one to seven (001-111). At the same time DWA and SDWA are initialized to use the less significant bits. In period 2, position data is modified by enabling the subset shifter at level 2. This splits the position data from (1-2) and (3-7) and individually rotated in the sense shown in Fig. 12. The accumulator provides modulo-7 addition of the thermometer inputs to generate new pointer values. Pointer (SS)/shift data is a modulo-7 sum of all the thermometer inputs. Since our input is constant 2 thermometer levels, Pointer (SS)/shift data increases two each succeeding clock period but after reaching seven it loops back to one. The position value enabled by the DWA data becomes the SDWA output.

3.2. Result analysis 1

3.2.1. SDWA randomization (DC input – 16 periods)

![Figure 11. SDWA Randomization (DC input, 16 periods).](image)

To show that SDWA provides randomization, we refer to Fig. 11. Looking at the figure, we label each bit in high with their respective bit placement. Collecting the numbers generated, we have: 12, 45, 37, 46, 27, 16, 35, 14, 23, 67, 25, 17, 45, 36, 24, 16. Given this 16-period sequence, it shows that elements are used without any particular pattern, or in other words, the usage is randomized.
3.2.2 SDWA Averaging (DC input – 16 periods)
To show that SDWA provides averaging, we refer to Fig. 12 and Fig. 13 at 16 cycles having DC input.

Figure 12. SDWA averaging (DC input).

Figure 13. SDWA averaging (DC input).

Figure 13 shows tabulated averaging obtained from Fig. 12, Generated Split-set DWA data (at 10pF load, 16 periods). At each period we count how many times the particular unit elements/bits are used for the 16-period sequence. As tabulated above, in every period, it is shown that the elements are equally used or differs only one bit from the other. A bit is not used twice unless all the others are used once. This characteristic is maintained throughout the whole period and thus for long term, we can say that it provides equal element usage.

3.2. SDWA simulation (time-varying thermometer input, 13 periods)

Figure 14.

Figure 14. SDWA simulation (with time-varying thermometer input).

Figure 15.

Figure 15. SDWA output (with time-varying thermometer input).

Figure 14 shows a digital time-varying thermometer input for 13 clock periods. Having this digital input, SDWA output is shown in Figure 15. It can be observed that averaging is done since each unit element is used six times. On the other hand, randomization is also present since there is no particular pattern in element assignment.

3.3. SDWA simulation at three major corners (TT, FF, and SS) with time-varying input

Figure 16.

Figure 16. Layout simulation at three major corners (TT, FF, SS) for 7 periods.
TT corner (red) is simulated at 25°C, with the normal supply level and using typical-typical Synopsys CMOS library. FF corner (blue) is simulated at 0°C, at the 1.1 of supply level and using fast-fast Synopsys CMOS library. SS corner (green) is simulated at 125°C, with 0.9 of the supply level and using the slow-slow Synopsys CMOS library. The three corner simulations provide the same output logic.

3.4. SDWA output simulation (schematic vs. layout)
Figure 17 shows the overlay mode of the schematic vs. layout output simulation. The graph verifies the logic equivalence of the schematic and the layout output. Graph colored in red is the schematic output simulation while the one in blue is the layout simulation. The pre-simulation and post-simulation as shown above verify the functionality of the design.

Figure 17. SDWA output simulation.

Figure 18. Output delay.

Figure 18 shows the delay between the schematic and layout output with the time-varying discrete input. The layout output simulation is delayed by 108 picoseconds in reference with the schematic simulation.

3.5. SDWA output simulation (with and without the output register)

Figure 19. SDWA simulation (without output register block).

Figure 20. SDWA simulation (using output register).

Due to the randomization scheme of SDWA, unmatched delays are also randomized. These timing problems are clearly illustrated in Fig. 19, where the output are distorted by glitches and jitters. These are inevitable to some extent because randomization results to unequal or unparallelled delay between processes. The problem could extend more in the actual layout and which is very realistic. The researchers wish to filter or minimize these problems by introducing an output register.

The output register will store the data before it will be used in the next block. Doing this, the significant distortion is eliminated or atleast minimized significantly. In consequence of using an output register, the input to output response will be delayed at least one cycle of the clock. In Fig. 20, it is shown how the output is made better since of lesser distortions.
3.6. SDWA output simulation (power consumption)
At typical NMOS-Typical PMOS corner (TT corner), the total power consumption is 566.348microwatts or approximately equal to 0.566 milliwatts at 50 Megahertz clock frequency.

Figure 21. SDWA zoomed layout.

4. References
[1] Sall E, Andersson K O and Vesterbacka M 2004 Proceedings of the 6th Nordic Signal Processing Symposium 137–40
[2] Pelgrom M J M, Tuinhout H P and Vertregt M 1998 International Electron Devices Meeting 915–8
[3] Wang R and Temes G C 2006 Electronics Letters 42 248–9
[4] Bruce J W 2000 Dynamic Element Matching Techniques for Data Converters (Las Vegas: University of Nevada)
[5] Wang R 2006 A Multi-Bit Delta Sigma Digital-to-Analog Converter Dissertation, Oregon State University, Oregon, US

Acknowledgment
This work was supported by Department of Science and Technology – ERDT Fund, Philippines and MSU-IIT College of Engineering, Micro-Electronics Laboratory.