DISP '19, Oxford, United Kingdom
ISBN: 978-1-912532-09-4

A 5.25ps-resolution TDC on FPGA using DSP blocks

Scott Tancock
School of Computer Science, Electronic Engineering
and Engineering Mathematics
University of Bristol
United Kingdom
0044117 3315326
scott.tancock@bristol.ac.uk

Naim Dahnoun
School of Computer Science, Electronic Engineering
and Engineering Mathematics
University of Bristol
United Kingdom
0044117 9545181
naim.dahnoun@bristol.ac.uk

ABSTRACT

1. With the introduction of its 7-series FPGAs, Xilinx introduced a new iteration of their DSP48 block, the DSP48E1. In our previous paper, we examined the usefulness of its predecessor, the DSP48A1, for the generation of high-resolution delays. In this paper, we examine how the DSP48E1 has changed and what this means for the generation of high-resolution delays with this block.

Keywords

Time-to-Digital Converters, Field-Programmable Gate Arrays, Xilinx, Delay Generation.

2. INTRODUCTION

Time-to-digital converters are major components in many industrial, scientific and medical applications, from the laser rangefinders present on emerging autonomous cars (LIDARs) [1] to the all-digital phase-locked loops [2] found in modern microprocessors. Fundamental to time-to-digital converters (TDCs) is the ability to reliably generate high-resolution delays with which to compare the signal. Modern methods on FPGAs utilise very simple delay lines created from the smallest on-chip components possible, most commonly the carry chains designed for fast ripple-carry addition [3].

In our previous paper [4], we investigated the feasibility of using the dedicated addition logic inside the DSP48A1 block present on 6-series FPGAs from Xilinx. We subsequently concluded that, due to the high variance in bin sizes caused by the carry look-ahead logic present in the adder, the DSP48A1 block was less useful than the carry chains for high-resolution delay generation, but showed promise as a semi-fine delay generator, as the 800ps per-DSP delay was relatively stable.

In this paper, we extend our previous work to examine the DSP48E1 blocks present in the 7-series FPGAs. The DSP48E1 blocks are a significant upgrade compared to the DSP48A1 blocks seen previously, implementing a pattern recogniser, deeper pipelining, higher clock speeds, SIMD and 2-input logic functions [5]. We once again utilise the adder in the DSP as this provides a fast carry through multiple elements and perform code density tests to determine the feasibility of using the new DSP blocks as delay generators.

The rest of this paper will be structured as follows: first, there will be a section describing the changes made to the DSP48E1, how they affect the design, and how the system is configured to obtain delays from these elements. Then, there will be a section detailing our testing methodology. After that, there is a section on the results obtained from our tests. Next is a section discussing the results obtained. Finally, there will be a conclusion section and discussion of further work.

3. SYSTEM DESIGN

3.1 Singular DSP

As our aim was to turn the 48-bit post-adder in the DPS48E1 into a delay line, we needed to examine its operation. The post-adder has three major inputs, the 48-bit X, Y and Z multiplexers (muxes), as well as 1 minor input (the carry input), one major output (the P output), 1 minor output (the carry out) and two control inputs, the operation mode and arithmetic logic unit (ALU) mode. In addition, there are several configuration registers to enable or bypass various pipeline registers in the DSP48E1.

As with the DSP48A1, all pipeline registers were disabled except for the P output register. In the DSP48A1, the carry cascade output was connected to a different register to the P output, so the P output register could be used as a discriminator while the carry register was bypassed. In the DSP48E1, the carry output and P output both go through the same register (different bits) so the register must be disabled to allow the carry to propagate to the next DSP asynchronously. Therefore, external fabric flip-flops were required to register the output.

Unlike the DSP48A1, the Y mux has an option to use a hard-coded 48 binary ones as the input, so we decided to use this rather than manually inputting the same pattern. Similarly, for the trigger, the DSP48E1 adds to the DSP48A1 by including a dedicated carry input from the general-purpose FPGA fabric, so we were able to use this for the east-significant DSP's carry input and have all DSPs with the X and Z inputting all zeros, again through a hard-wired input to the mux.

The operation mode input was set to addition, while the ALU mode was set to add all inputs together. The carry selection logic was set to use the FPGA carry input for the least-significant DSP and the cascaded carry input for subsequent DSPs.

3.2 Multiple DSPs And TDC Design

Once we had determined the set-up for a single DSP, we then sequenced 20 of them to be sure we would cover the whole clock period, resulting in a maximum of 960 bins. These bins were registered by two levels of flip-flops on nearby FPGA fabric (to eliminate metastability), and then inverted and consumed by a priority encoder, which determines the most significant position of a one (a zero before inversion) to see how far the delay propagated. These most significant positions were then output to a PC via a parallel interface (Digilent's DPTI) for histogramming.

3.3 Population Counter

Due to the disappointing results of the histogram (multiple missing bins, see Section 5), we determined that we needed a way to recover these missing bins. The missing bins occur due to the flip-flop on bin i+1 passing its blackout time before the flip-flop in bin i,

DOI: http://dx.doi.org/10.17501.......................................

DISP '19, Oxford, United Kingdom
ISBN: 978-1-912532-09-4

meaning the priority encoder does not pick up the transition of bin i since bin $i + 1$ is already high.

As the number of bins that are high is monotonically increasing, we can instead use a population counter for un-ordered bin detection. With a population counter, even if bin i transitions after bin $i + 1$, the two transitions will still give two different outputs, separated by a single LSB. With this, we were able to recover the missing bins, although these bins were extremely small.

3.4 Parallel Delay Lines

Although the bins were recovered, there was still a significant difference in size between the last bin in a DSP and the rest. As the sum of the rest of the bins was more than 1/3 of the size of the large bin (170ps vs 310ps), it was possible to sub-divide the large bins by creating multiple delay lines with fractions of a DSP as an offset, and then summing the results of these delay lines. As each delay line is monotonically increasing, stepwise, as the time between the trigger and clock increases, the sum of the delay lines must also monotonically increase, with each delay-line sub-dividing the others' large bins.

The offsets, being much smaller than a DSP on its own, were provided by CARRY4 blocks. As each CARRY4 block is, on average, 65ps and the delay of a DSP block is 550ps, two CARRY4 blocks are sufficient to provide the first delay, with subsequent delays successively adding pairs of CARRY4 blocks to generate successive offsets.

4. METHODOLOGY

For the DSP48E1, we used a linear code density test to determine the delay of each component. The results of the code density test were then histogrammed per-DSP and per-bin to determine the overall delay of each DSP block in the chain as well as the bin distribution within a DSP. We used a 1MHz external pulse generator to generate the rising edges (start signals) asynchronously to the system clock (stop signal) to obtain an even distribution of delays with respect to the system clock, which had a frequency of 120MHz. Read-out was performed live for 1,041,043 tags and histogramming performed off-line.

5. RESULTS

5.1 Per-DSP

Figure 1 shows the time bins on a per-DSP basis. When discarding outliers, we observe a 553ps mean delay with 74ps standard deviation. DSP 0 is an outlier that describes the path mismatch between the trigger / coarse counter and the fine counter. DSPs with indexes greater than 14 are completely outside the clock period and only occur due to metastability in the triggering logic, while DSP 14 is only partially (25%) within the clock period and so can be considered an outlier.

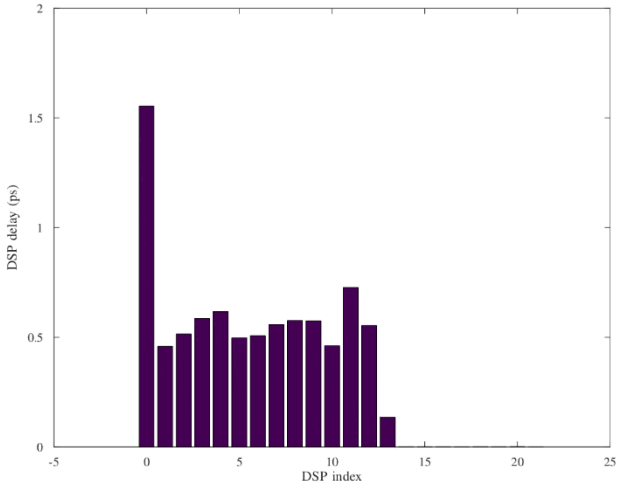

Figure 1: Delays of the DSP blocks.

5.2 Per-bin

Figure 2 shows the time bins on a per-bin basis, averaged across all non-outlier DSPs. We see that most of the delay is concentrated in the final bin (bin 47), while the rest is scattered across the bins approximately evenly. However, due to the high final bin, the mean is 11.5ps and the standard deviation 45.1ps (would be 5.17ps and 9.72ps respectively if not for the final bin).

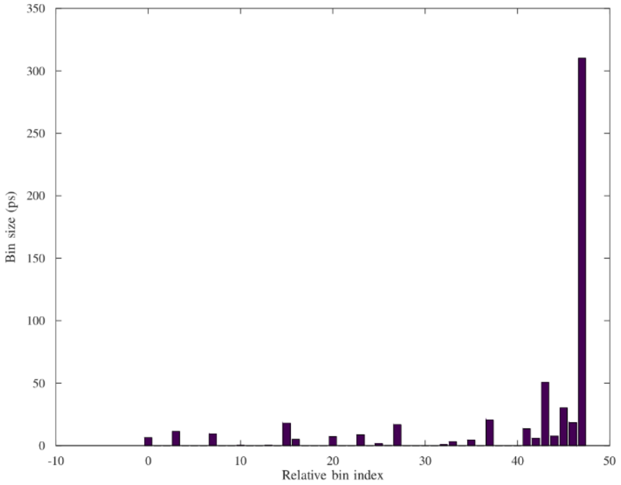

Figure 2: Average delay within a DSP block.

5.3 Overall

Figure 3 shows the time bins across the DSP blocks. All the large peaks occur in the last bin (47) of a DSP, demonstrating the repeatability of the results across multiple DSPs.

DOI: http://dx.doi.org/10.17501.......................................

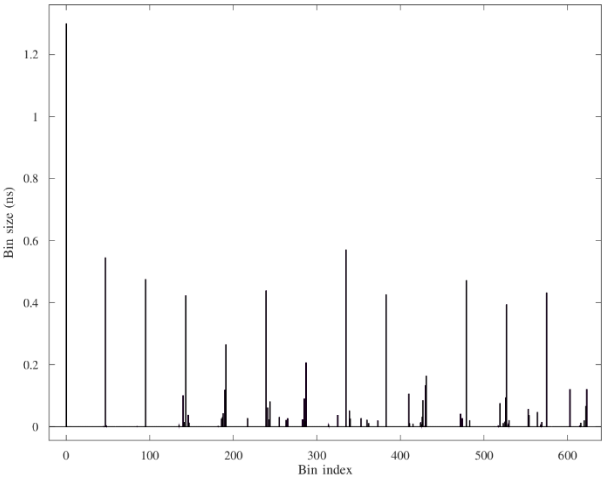

Figure 3: Linear code density test.

5.4 With Population Counter

Figure 4 shows the whole-system histogram when the population counter is used with a single delay line. Unlike the results shown in Figure 2, it shows that each bin other than the large one at the end has some hits in it, corresponding to approximately 5.21ps per bin. The large final bin still exists.

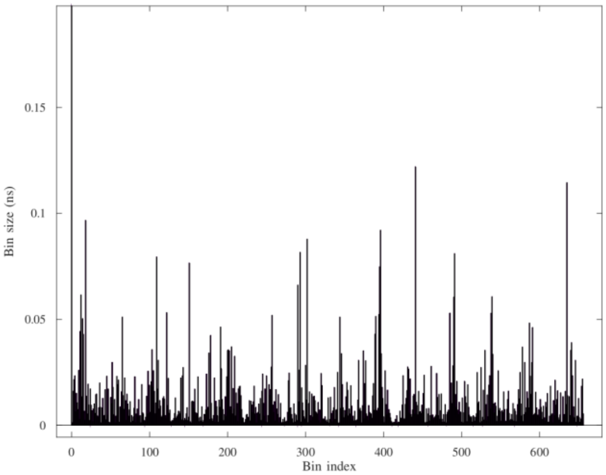

Figure 4: Linear code density test with population counter. Bin 0 is 447ps.

5.5 Parallel Summed Delay Lines

Figure 5 shows the whole-system histogram when the population count outputs of 4 offset delay lines are summed together. Worthy of note in this diagram is that the maximum bin number is quadruple that of a single delay line (as we are summing 4 delay lines) and that the large bins present in the previous tests no longer exist. This occurs because one delay line is propagating through its small (5.21ps) bins while the others propagate through their large bins, resulting in the summed output code still being as finely divided as the delay line propagating through the small bins.

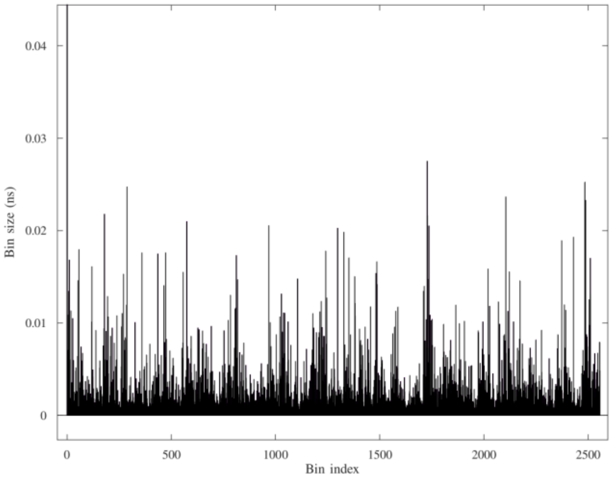

Figure 5: Code density test with 4 offset delay lines. Bins 0 and 1 are 710ps and 106ps respectively.

6. DISCUSSION

Similarly to the results seen on the DSP48A1 blocks, the DSP48E1 blocks suffer from both missing bins and large non-linearities when used with a priority encoder. As we can be sure that the outputs are monotonically increasing, we can instead use a population counter, and this removes the issue of missing bins. This creates a delay line with a large delay of (on average) 308ps, with the other 245ps rest of the delay split between 47 bins for an average of 5.21ps per bin.

To sub-divide these large bins, an equivalent coding line [6] can be created by summing together the outputs of multiple DSP delay lines. As the large bin is many times larger than the average bin or the standard deviation, we cannot rely on natural drift due to process variation, and so must insert delays to offset the DSP delay lines relative to each other. This is done by inserting varying numbers of CARRY4 chains before the delay lines to form the correct delay duration.

Once 4 DSP delay lines have been successively offset by a quarter of the duration of a delay line each, their outputs are summed to form an equivalent coding line (ECL) of 2557 bins length. This ECL has a max bin size of 27.04ps and a cubic mean bin size of 5.25ps, with the first two bins discarded due to system offsets (which can be compensated for). 5.25ps is an acceptable resolution for the quantity of logic (9% of the system's DSP blocks) required to implement it and so this can be considered a successful implementation. Further testing revealed that the system's maximum bin size can be dropped to 22.35ps with a cubic mean of 3.70ps using 18% of the system's DSP blocks.

7. CONCLUSION AND FURTHER WORK

In conclusion, the DSP blocks present in the Artix-7 Series FPGAs are highly non-linear (more than half the delay is contained within a single bin) and suffer from out-of-order bins (which show up as missing bins with a priority encoder). By introducing a population counter to convert the output code, the out-of-order bins were rectified, and by offsetting 4 parallel delay lines (9% DSP resources), the large bins were sub-divided through the equivalent coding line method. This produced a 5.25ps bin resolution with 27.04ps max bin size and doubling the number of parallel delay lines to 8 (18% DSP resources) produced 3.70ps resolution with a 22.35ps maximum bin size.

In the future, we intend to analyse the DSP blocks on competitor chipsets (e.g. Intel/Altera) and incorporate these delay lines in larger systems and various applications such as laser range-finding and quantum photonics experiments. Using higher-end FPGAs from the Xilinx 7-Series or Ultrascale(+) series, it is possible to utilise higher-grade logic components with smaller delays (hence better resolution), as well as more logic in parallel. With this higher-grade logic, it is expected that sub-ps level resolution can be achieved in single-channel operation.

8. REFERENCES

[1] Y. Cao, W. De Cock, M. Steyaert and P. Leroux, “Design and Assessment of a 6 ps-Resolution Time-to-Digital Converter With 5 MGy Gamma-Dose Tolerance for LIDAR Application,” *IEEE Transactions on Nuclear Science,* vol. 59, no. 4, pp. 1382-1389, 2012.

[2] R. B. Staszewski, K. Muhammad, D. Leipold, C.-M. Hung, Y.-C. Ho, J. L. Wallberg, C. Fernando, K. Maggio, R. Staszewski, T. Jung, J. Koh, S. John, I. Y. Deng, V. Sarda, O. Moreira-Tamayo, V. Mayega, R. Katz, O. Friedman, O. E. Eliezer, E. de-Obaldia and P. T. Balsara, “All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS,” *IEEE Journal of Solid-State Circuits,* vol. 39, no. 12, pp. 2278-2291, 2004.

[3] P. Chen, H. C. Cheng, A. Widodo and W. X. Tsai, “A PVT insensitive field programmable gate array time-to-digital converter,” in *2013 IEEE Nordic-Mediterranean Workshop on Time-to-Digital Converters (NoMe TDC),* Perugia, 2013.

[4] S. Tancock, S. Mehmood, E. Arabul and N. Dahnoun, “Can DSP48A1 adders be used for high-resolution delay generation?,” in *The 7th Mediterranean Conference on Embedded Computing - MECO'2018,* Budva, 2018.

[5] Xilinx, “7 Series DSP48E1 Slice User Guide,” 27 March 2018. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug479_7Series_DSP48E1.pdf. [Accessed 21 October 2018].

[6] R. Szplet, Z. Jachna, P. Kwiatkowski and K. Rozyc, “A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines,” *Measurement Science and Technology,* vol. 24, no. 3, p. 035904, 2013.

DOI: http://dx.doi.org/10.17501.......................................