Design and Implementation of a Re-configurable Versatile Direct Digital Synthesis based Pulse Generator

Ashok Sharma, Yichuang Sun *Senior Member, IEEE* and Oluyomi Simpson *Member, IEEE*

***Abstract*— Direct digital synthesis (DDS) technology has replaced many traditional analogue circuit functions. The frequency, amplitude and phase of a waveform generated using DDS can be precisely controlled. It is possible to achieve remarkably high frequency resolution, fast frequency switching, and modulation can be easily introduced in a DDS generator. However, pulse waveform generated using DDS requires re-loading of the pulse waveform in the waveform memory to vary the pulse parameters which causes phase discontinuity, requires prohibitively large memory to generate pulses with narrow widths and long periods and suffers from trigger uncertainty of up to one clock period when the pulse waveform is triggered by an external event. Moreover, it is not possible to modulate the width and delay of the pulse stored in the waveform memory. This paper describes a fully functional, low complexity, low cost, memory less, parallel and DDS based pulse waveform generator implemented in a low-cost field programmable gate arrays (FPGA) where pulse parameters such as amplitude, period, width, rise time, fall time and delay can all be varied without any glitches or phase discontinuity, where the amplitude, frequency and phase, as well as period, width and delay of the pulse can be modulated by an internally generated or external modulating signal and where it is possible to trigger the generation of the pulse or burst of pulses by an external event where the time between the trigger event and the pulse output is fixed.**

***Index Terms*— Direct Digital Synthesis (DDS); Field Programmable Gate Arrays (FPGA); Pulse Width Modulation (PWM); Re-configurable**

I. Introduction

Pulse generator is an important test and measurement instrument. High precision, high frequency, fast edge, modulated pulses find application in many fields including semiconductor testing for characterising devices and materials, testing collision avoidance in automotive, simulation of sensor signals, testing of wideband in-phase and quadrature modulation and power amplifier testing. They are also used for performance verification in the field of radiography, mass spectroscopy, laser and military radar, electromagnetism and in general research and development [1] [2].

Direct digital synthesis (DDS) first proposed by Tierney et al in 1976 is a popular method that is used to generate waveforms of any shape that has a linear and periodic phase [3]. Figure 1 is a simplified block diagram of a DDS.

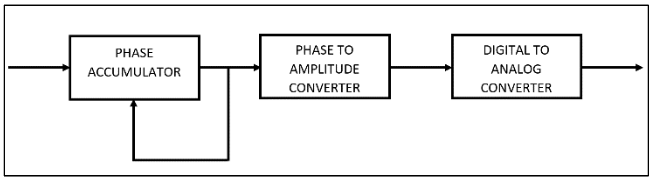


Figure 1. Simplified block diagram of a DDS.

The DDS system could be used to generate pulse waveform [4]. In its simplest form, the most significant bit of the phase control word output from the phase accumulator is a pulse waveform with zero delay, minimum fixed rise and fall times and equal mark to space ratio at the frequency of the DDS [5]. However, it is prone to jitter. If the frequency control word input to the phase accumulator divides into 2^N where N is the number of bits in the phase accumulator, the output is periodic and smooth, but all other cases will create jitter. The jitter can vary up to 1 clock period, is fully deterministic and could be reduced using a delay generator [5]. However, a delay generator requires a lot of hardware and would have been difficult to implement for a single bit pulse output and even more difficult for multiple bits.

It is also possible to generate pulse waveform by counting clock cycles [6]. The clock used could be a DDS generated clock, a digital sine waveform generated using DDS, converted to analogue using digital to analogue converter (DAC) and the re-construction filter and then converted into square waveform clock using a comparator, to have the ability to slightly vary the clock frequency to increase pulse period resolution. Clock cycles are counted to determine period and pulse width. Edge times could be controlled by a pulse edge shaping circuit in the analogue domain. The drawback of this method is that the resolution of the pulse width is limited to the clock period. The rising and falling edge times could be controlled by a circuit that varies the charging currents in a capacitor. However, such a circuit would only offer limited control of the edge times,

This research was supported by UK Technology Strategy Board (now UK Research and Innovation - Innovate UK) and Thurlby Thandar Instruments Limited under Grant KTP006473 and SKTP1000363.

Ashok Sharma is with Thurlby Thandar Instruments Limited, 2 Glebe Road, Huntingdon, Cambridgeshire PE29 7DR.

Yichuang Sun and Oluyomi Simpson are with the School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, Herts AL10 9AB, United Kingdom. (Corresponding Author, Yichuang Sun, Email: y.sun@herts.ac.uk).

usually from a few nanoseconds to a few microseconds in limited ranges.

The focus of this research has been to achieve an entirely digital DDS-based pulse generator. The output of the phase accumulator is used to address a memory which contains the desired waveform shape. To generate a pulse waveform, a pulse shape with the desired delay, rise time, width and fall time could be downloaded in the waveform memory. However, to change any of these parameters, re-loading of the memory would be required which will cause glitches or phase discontinuity at the output. For a given period, the size of the memory will limit the shortest width or edge time that could be achieved. For example, if the waveform memory is 65536 words long and if the DDS output frequency is set to 1 Hz, pulse period would be 1 second and it would not be possible to define the pulse width or edge times to less than 15 micro-second.

Figure 2 shows how modulation can be achieved in a DDS system [5]. The DDS waveform can be frequency modulated by adding the modulating input to the frequency control word input of the phase accumulator [7]. Adding the modulating input to the phase control word output of the phase accumulator before it addresses the phase to amplitude converter results in phase modulation [8]. Multiplying the digital waveform with the modulating signal before it is converted to its analogue form modulates the amplitude of the DDS waveform [9]. Therefore, it would be possible to achieve, frequency, phase and amplitude modulation for a pulse waveform stored in the memory. However, it would not be possible to achieve pulse delay and pulse width modulation. It is also desirable sometimes to modulate the period of the pulse waveform without affecting the pulse delay, width, and edge times. This also would not be possible.

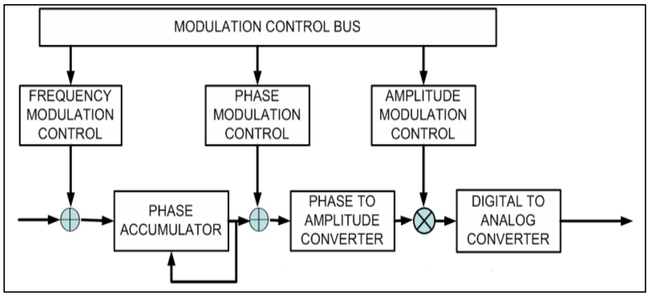

MODULATION CONTROL BUS
FREQUENCY MODULATION CONTROL
PHASE MODULATION CONTROL
AMPLITUDE MODULATION CONTROL
PHASE ACCUMULATOR
PHASE TO AMPLITUDE CONVERTER
DIGITAL TO ANALOG CONVERTER

Figure 2. Simplified DDS architecture with modulation capability.

This paper proposes adaptation of the DDS system for digital pulse generation using mathematical means without using any memory. Pulse parameters can be changed instantaneously without any glitches. It is possible to generate pulses with the narrowest width and widest period. The novel architecture makes it extremely easy to introduce pulse period, width, and delay modulations. The paper also proposes using this new DDS based design in parallel to increase the maximum frequency of the pulse waveform. Finally, the paper shows how jitter is reduced in this new architecture.

A DDS system can output a specified number of waveforms, known as burst, by counting the number of times an overflow occurs in the phase accumulator. The accumulator is turned off when the count becomes equal to the specified number of cycles [10]. This starting and stopping of the phase accumulator normally happens after a trigger event [10, 11] . In case of an external trigger event [12], the generation of waveform begins on the next DDS clock after the event, which introduces a time uncertainty of up to 1 clock period. This is not acceptable in many pulse applications.

If the time between the start of the trigger and the start of the subsequent DDS clock could be measured, then this information could be used to offset the phase accumulator before starting the waveform to achieve a constant latency between the trigger and the waveform output. This paper demonstrates how this could be achieved in the proposed architecture. It should be declared that this paper, reuses some content from thesis [13] with permission.

Section II of this paper provides the system model, methodology and details the design analysis of the proposed pulse generation architecture. Section III details the field programmable gate arrays (FPGA) design, synthesis, and results of the new method. Section IV concludes the research.

II. RE-CONFIGURABLE PULSE GENERATOR DESIGN

A. Pulse Generator

The output from the phase accumulator, which is a ramp waveform could be directly used to generate a pulse waveform which is a waveform made up of five straight lines of varying slopes. Figure 3 shows the relation described above. The output from the phase accumulator increments from 0 to $2^N - 1$, where N is the number of bits in the phase accumulator, in increments of the frequency control word input at the clock rate. When the phase accumulator reaches or gets close to its maximum value, it overflows and starts accumulating again. For a pulse waveform, *delay* is the time for which the pulse remains low before it starts to rise, with respect to a trigger event or with respect to the start sync, *rise* is the transition time of the pulse from low level to high level measured between 10% and 90% of the threshold points, *width* is the time from the 50% threshold of the rising edge to the 50% threshold of the falling edge of the pulse, and *fall* is the transition time of the pulse from high level to low level measured between 10% and 90% of the threshold points. The transition points A, B, C and D as shown in Figure 3 could be represented in terms of phase output as follows.

$$A = \frac{(delay \times 2^N)}{period} \tag{1}$$

$$B = \frac{(delay + (1.25 \times rise)) \times 2^N}{period} \tag{2}$$

$$C = \frac{(delay + (0.625 \times rise) + width - (0.625 \times fall)) \times 2^N}{period} \tag{3}$$

$$D = \frac{(delay + (0.625 \times rise) + width + (0.625 \times fall)) \times 2^N}{period} \tag{4}$$

Two other values, E and F are defined as follows.

$$E = \frac{period}{(1.25 \times rise)} \tag{5}$$

$$F = \frac{period}{(1.25 \times fall)} \tag{6}$$

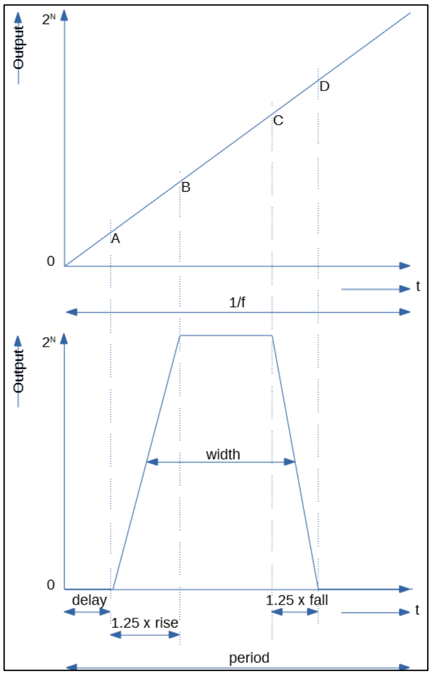


Figure 3. Relation between DDS phase accumulator output and desired pulse shape.

Four N-bit comparators would be needed to compare the phase accumulator output value, P, with A, B, C and D respectively. There are three results from this comparison block, $M1$, $M2$, and S. Table 1 shows the relation between the inputs and outputs of the comparison block.

$M1$ and $M2$, both N bit values are inputs to a multiplier. The output of the multiplier would be $2 \times N$ bits long. Depending on the DAC resolution, only M most significant bits of the multiplier output is passed to the 'inverse' block that follows the multiplier, where M is the number of input bits to the DAC. Due to the large number of bits, the multiplier may have to be pipelined and is performed over a few clock cycles by adding registers in the intermediate stages. The single bit S value is delayed by the same number of clock cycles it takes to perform the multiplication and passed to the 'inverse' block. In the 'inverse' block, if the delayed S value is equal to 0, then the $output$ is equal to the $input$. If the delayed S value is equal to 1, then $output$ is equal to $2^M - 1 - input$. The $output$ of the 'inverse' block is the desired pulse waveform. Figure 4 shows the main building blocks of the pulse generator.

Table 1. Comparison block – input / output relation.

Conditions	M1	M2	S
$P \le A$	0	0	0
$A < P \le B$	E	$P - A$	0
$B < P \le C$	0	0	1
$C < P \le D$	F	$P - C$	1
$D < P$	0	0	0

Figure 5 is a simplified block diagram of the proposed DDS pulse generator. N bit values , A, B, C and D, E, F and the N bit frequency control word value input to the phase accumulator are set by the controlling processor, based on the user desired pulse *period*, *delay*, *width*, *rise* and *fall* times respectively. If the individual control words are set immediately, there is a potential of setting incorrect pulse parameters momentarily while the processor is updating the control words. This could be avoided by having two sets of registers, namely buffer registers, and active registers. The processor updates the buffer registers, and the active registers are only updated when all buffer registers have been updated. This allows for glitch-free and phase continuous parametric changes. These seven control values completely define the pulse waveform and therefore it is possible, if it is so desired, to set them such that $period$ and $width$ are changed simultaneously to maintain a constant duty cycle, $width$, $rise$ and $fall$ times are changed simultaneously to maintain a constant pulse shape, delaying only

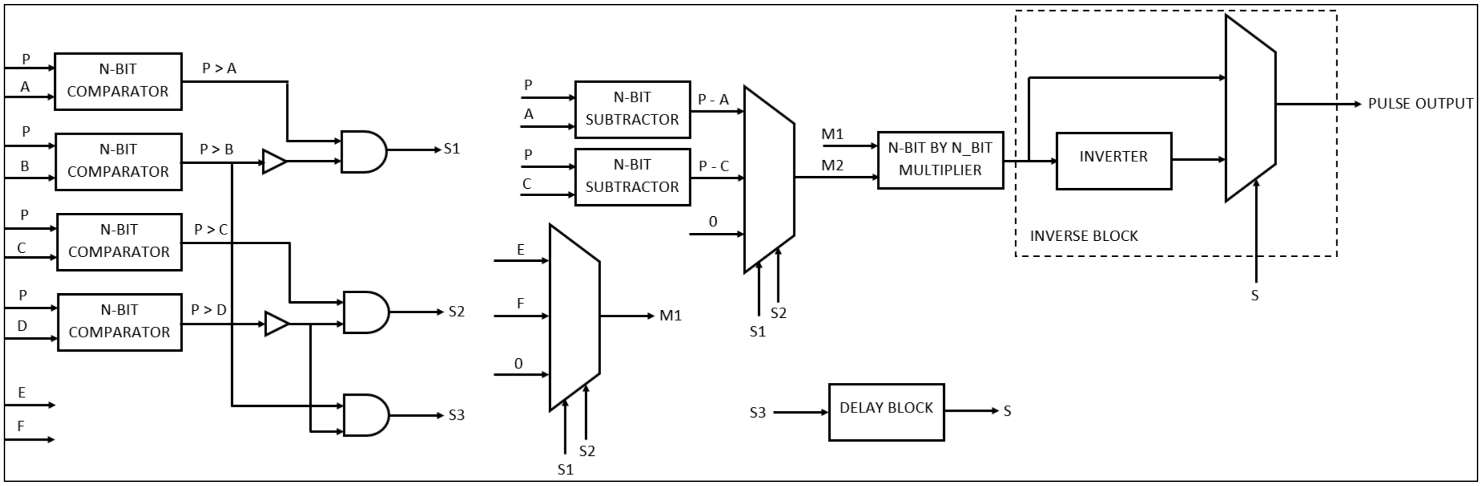


Figure 4. Simplified block diagram of the pulse generator.

the falling edge, and any other combination that might be useful for the end application, all in a phase continuous manner.

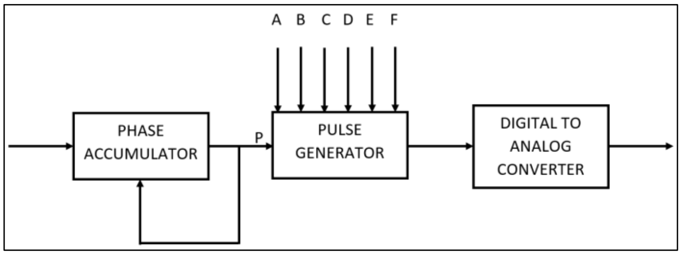


Figure 5. Simplified block diagram of DDS Pulse Generator.

For any pulse waveform, there are some fundamental limitations. They are as follows:

$$period \geq delay + (0.625 \times rise) + width + (0.625 \times fall) \quad (7)$$

$$width \geq (0.625 \times rise) + (0.625 \times fall) \quad (8)$$

For the pulse waveform to be free of any timing jitter, there should be at least one waveform point on the high level and one waveform point on the low level. Five waveform points on the rising and falling edges guarantee the desired $rise$ and $fall$ times and removes timing jitter from the edges. The limitation of the pulse waveform with the new architecture are as follows:

$$\begin{aligned} period \geq\ & delay + (0.625 \times rise) + width \\ & + (0.625 \times fall) + \left(\frac{1}{fclk}\right) \end{aligned} \quad (9)$$

$$width \geq (0.625 \times rise) + (0.625 \times fall) + \left(\frac{1}{fclk}\right) \quad (10)$$

$$(1.25 \times rise) \geq \frac{5}{fclk} \quad (11)$$

$$(1.25 \times fall) \geq \frac{5}{fclk} \quad (12)$$

If the clock frequency is 200 MHz, the minimum $rise / fall$ times can be calculated from equations (11) and (12) as follows.

$$RISE_{MIN} = FALL_{MIN} = \frac{5}{(1.25 \times fclk)} = 20ns \quad (13)$$

Minimum $width$ can then be calculated from equation (10) as follows.

$$\begin{aligned} WIDTH_{MIN} &= (0.625 \times RISE_{MIN}) + (0.625 \times FALL_{MIN}) \\ & + \left(\frac{1}{fclk}\right) = 30ns \end{aligned} \quad (14)$$

Minimum $delay$ could be set to 0 second. Minimum $period$ can then be calculated from equation (9) as follows.

$$\begin{aligned} PERIOD_{MIN} &= DELAY_{MIN} + WIDTH_{MIN} + (0.625 \times RISE_{MIN}) + \\ & (0.625 \times FALL_{MIN}) + \left(\frac{1}{fclk}\right) = 60ns \end{aligned} \quad (15)$$

If the number of bits in the phase accumulator is 112, then the minimum frequency, and therefore, maximum $period$ can be calculated from the DDS principles [5] as follows.

$$\begin{aligned} F_{MIN} &= \frac{fclk}{2^N} = 3.8e^{-26} Hz \\ PERIOD_{MAX} &= \frac{1}{F_{MIN}} = \frac{1}{3.8e^{-26}} = 2.5e^{24}\ \mathrm{s} \end{aligned} \quad (16)$$

Maximum $period$ is restricted to 1000000 seconds as there are very few applications that require pulse $period$ greater than a thousand seconds. Maximum $width$ can be calculated from equation (9) as follows.

$$\begin{aligned} WIDTH_{MAX} &= PERIOD_{MAX} - DELAY_{MIN} - (0.625 \times RISE_{MIN}) \\ & - (0.625 \times FALL_{MIN}) - \left(\frac{1}{fclk}\right) = 999999.999999970ns \end{aligned} \quad (17)$$

Maximum $delay$ can be calculated from equation (9) as follows.

$$\begin{aligned} DELAY_{MAX} &= PERIOD_{MAX} - WIDTH_{MIN} - (0.625 \times RISE_{MIN}) \\ & - (0.625 \times FALL_{MIN}) - \left(\frac{1}{fclk}\right) = 999999.999999940ns \end{aligned} \quad (18)$$

Maximum $rise$ time can be calculated from equations (9) and (10) as follows.

$$\begin{aligned} WIDTH_{RISEMAX} &= (0.625 \times RISE_{MAX}) + (0.625 \times FALL_{MIN}) + \left(\frac{1}{fclk}\right) \\ PERIOD_{MAX} &= DELAY_{MIN} + WIDTH_{RISEMAX} + (0.625 \times RISE_{MAX}) \\ & + (0.625 \times FALL_{MIN}) + \left(\frac{1}{fclk}\right) \\ PERIOD_{MAX} &= DELAY_{MIN} + (1.25 \times RISE_{MAX}) \\ & + (1.25 \times FALL_{MIN}) + \left(\frac{2}{fclk}\right) \\ RISE_{MAX} &= \frac{(PERIOD_{MAX} - DELAY_{MIN} - (1.25 \times FALL_{MIN}) - (\frac{2}{fclk}))}{1.25} \\ &= 799999.999999972s \end{aligned} \quad (19)$$

Similarly, maximum $fall$ time will also be 799999.999999972 seconds.

In a DDS architecture, the resolution of $period$ decreases with increase in $period$ value. With a clock frequency of 200 MHz and 112 bits in the phase accumulator, for the maximum $period$ of 1000000 seconds or 1 μHz frequency, the frequency control word input to the phase accumulator can be calculated as follows.

$$FCW = \frac{(2^N \times fout)}{fclk} = 25961484292674138142 \quad (20)$$

If the $period$ is now decremented by 10 pico-second, the frequency control word will be,

$$FCW = \frac{(2^N \times fout)}{fclk} = 25961484292674397757 \quad (21)$$

This shows that even at the maximum $period$ end, there is enough resolution in the phase accumulator to achieve better than 10 pico-second resolution in the $period$ setting. Pipelined phase accumulator architecture could be used to design a phase accumulator with large number of bits [14]. The output of the phase accumulator is truncated to 48 bits before being passed to the pulse generator architecture. The $delay$, $width$, and $rise/fall$ times are set by the 48-bit A, B, C, D, E, and F values and the $period$ is represented by 2^{48} which is equal to 281474976710656. This means that if the $period$ is less than or equal to 2814.74976710656 seconds, the architecture can achieve better than 10 pico-second resolution in the setting of $delay$, $width$, and $rise/fall$ times. If the phase accumulator output was truncated to 58 bits instead of 48 bits, then the architecture could achieve better than 10 pico-second resolution in the setting of $delay$, $width$, and $rise/fall$ times, for all $period$ settings. However, it is highly unlikely that 10 pico-second resolution would be required in any application in the setting of $delay$, $width$, and $rise/fall$ times when the $period$ is greater than 1000 seconds. The resolution in the pulse parameter settings will also be affected by the reference clock accuracy, DAC resolution, DAC non-linearity and the re-construction filter design that follows the DAC output.

With the proposed architecture, for a 200 MHz clock frequency, the minimum $period$ would be 60 nano-second, minimum $width$ would be 30 nano-second, and minimum $rise/fall$ times would be 20 nano-second. This may not be enough for some applications. One way to improve on this would be to operate four such generators in parallel [10, 15]. The output of the phase accumulator is added to 0, $Fclk/4$, $Fclk/2$ and $3 \times Fclk/4$, respectively to create four phase shifted inputs for the four parallel pulse generators. The other inputs A, B, C, D, E, and F are calculated in the usual manner. The outputs from the pulse generators are then multiplexed to create a single output at four times the clock rate. This multiplexing is usually done using dedicated hardware in the FPGA, explained in the next section. With these modifications, the minimum $period$ is decreased to 15 nano-second, minimum $width$ is decreased to 7.5 nano-second and minimum $rise/fall$ times is decreased to 5 nano-second. Subject to availability of resources, maximum data rate output from the FPGA, and the maximum sampling rate of the DAC, $period$, $width$, and $rise/fall$ times could be further reduced using eight or even sixteen parallel pulse generators.

B. Modulation

For modulation of the pulse waveform, the principle of DDS can be used to generate the internal modulating waveform [16]. Only phase accumulator and phase to amplitude converter blocks are required.

For a clock frequency of 200 MHz, 48 bits in the phase accumulator would achieve a frequency resolution of less than 1 μHz in the modulating waveform. 14 most significant bits of the phase accumulator outputs could be used to address a waveform memory containing 16384 16-bit waveform data. This would result in a memory size of 262144 bits which could easily be accommodated in the embedded memory of any low-cost FPGA. A waveform memory is used to allow the capability of modulating the pulse waveform with a waveform of any shape. The output from the memory block is the digital modulating waveform. This is then multiplied by a factor, amplitude depth in case of amplitude modulation (AM), frequency deviation in case of frequency modulation (FM), and phase deviation in case of phase modulation (PM), to have better modulation control. The amplitude depth, frequency deviation or pulse deviation value is set by the controlling processor. The scaled digital waveform is then either added to the frequency control word input to perform FM, added to the phase control word output to perform PM, or multiplied with the output of the pulse generator block to perform AM. If the scaled digital waveform is added or subtracted to the user specified amplitude value before being multiplied to the pulse waveform output, then normal AM is achieved. In this variant, when modulating waveform amplitude is at its minimum, the pulse output is zero. If the scaled digital waveform directly controls the amplitude of the pulse waveform output, then suppressed carrier AM is achieved. In this variant, when modulating waveform is zero, the pulse output is zero. It is also possible to add the scaled digital waveform to the output of the pulse generator to perform summing of the modulating waveform to the pulse waveform.

To perform pulse width modulation (PWM), the modulating waveform output from the memory is multiplied by the pulse width deviation factor and then added to C and D in the pulse generator architecture. To perform pulse delay modulation (PDM), the modulating waveform output from the memory is multiplied by the pulse delay deviation factor and then added to A, B, C and D in the pulse generator architecture. Pulse period modulation is explained in the next sub-section. Figure 6 shows the various modulations.

The multiplication block performs multiplication of the 16-bit modulating waveform output from the memory with a 48-bit control word resulting in a 64-bit output. This output is then truncated to appropriate number of bits for different types of modulations. A solution to large multiplication is presented in [17] where the first stage performs partial multiplications, the outputs of which are shifted and added in pipeline stages. The 48-bit control word is divided in three 16-bit control words and each is multiplied with the 16-bit modulating waveform. The results from each multiplication are shifted and then added together to produce the result. The output of the multiplier is passed on to various registers. Each type of modulation has its own synchronous reset signal to control the output from its corresponding register. When a modulation is enabled, its register outputs the multiplier output. When disabled, output from the register is zero.

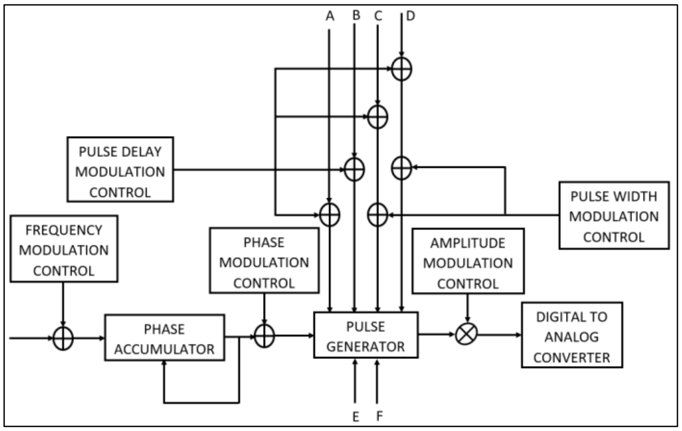

A
B
C
D
PULSE DELAY MODULATION CONTROL
PULSE WIDTH MODULATION CONTROL
FREQUENCY MODULATION CONTROL
PHASE MODULATION CONTROL
AMPLITUDE MODULATION CONTROL
PHASE ACCUMULATOR
PULSE GENERATOR
DIGITAL TO ANALOG CONVERTER
E
F

Figure 6. Simplified DDS pulse generator architecture with modulation capability.

For external modulation, the external modulating waveform is first converted to digital form by using an analogue to digital converter (ADC). It is possible to drive the external modulation ADC with the 200 MHz DDS Clock and have 16 bits of resolution such that the ADC output can directly be passed to the multiplication block as a modulating waveform. However, this is not a cost-effective solution. The ADC is sampled at a lower rate with fewer bits and therefore some sampling rate conversion and or interpolation is required. Interpolation method described in [18] could be used in this circumstance. If the ADC sampling rate is chosen to be some binary division of the DDS clock, then the difference between two consecutive A/D output samples could be divided by the same binary factor and then added to the previous sample at every DDS clock until a new sample is available. A binary factor is chosen to make the division a simple shifting operation. Cascaded integrator-comb (CIC) filters described in [19] are widely used in communication systems for efficient sample rate conversions and could also be used to up convert the sampling rate of the ADC samples to the DDS clock. The key advantage of CIC filters is that they use only adders and registers for any rate change, and do not require multipliers to implement in hardware for handling large rate changes. However, as the rate increases, the bit growth in the CIC filter increases as well. For a third order CIC filter with differential delay of one, a rate conversion of 16 adds 12 bits to the output.

C. *Trigger to Output Certainty*

It is possible to start and stop the phase accumulator to generate a specified number of pulse waveforms on receiving a trigger. In case of an external trigger event, the generation of waveform begins on the next DDS clock after the event which introduces a time uncertainty of up to 1 clock period, 5 nano-second for a 200 MHz clock frequency.

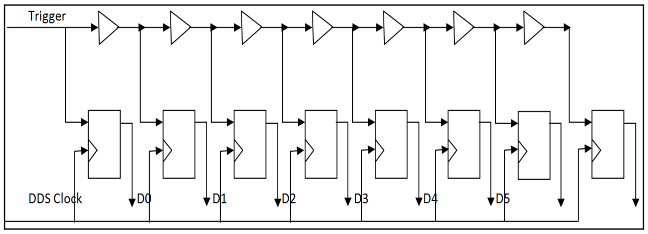


Figure 7. Delay line for trigger uncertainty compensation.

A tapped delay line shown in Figure 7 could be used to measure the time between the start of the trigger and the next DDS clock rising or falling edge as described in [20]. The input to the delay line is the trigger input and the output from each delay tap is clocked by the DDS clock. During static condition, all output bits are low. When the trigger arrives, some of the output bits will change its state from ‘low to high until the delay is long enough for the rising edge of the trigger signal to move beyond the rising edge of the clock signal after which point the outputs will remain low. The number of output bits that has changed state is an indication of the time between the trigger and the clock edge. As the trigger is delayed, it gets closer to the clock edge and registering it might suffer from violation of set up times resulting in meta-stability [21]. Therefore, two or more synchronous register stages should be used instead of one to ensure that the output is stable and that the probability of an unstable output due to meta-stability is infinitesimally small.

In many low-cost FPGAs available today, there exist chain structures or carry chains that the vendors design for general purpose applications. These chain structures provide short and predefined routes between identical logic elements. They are ideal for time to digital converter delay chain implementation as described in [12]. Figure 8 shows how the carry-in and carry-out chain available in most low-cost FPGAs could be used to implement a delay line for measuring the time between the start of the trigger and the next DDS clock. The logic elements used for the delay chain and the corresponding register array structure must be placed and routed by the FPGA compiler in a predictable manner to assure uniformity and short-term stability. The delay of the carry chain is subjected to variation due to temperature and power supply voltage and therefore needs compensation. The delay of the carry chain between two logic elements in the same logic array block (LAB) is different from the delay of the carry chain between two logic elements in two adjacent LABs. If the delay line exceeds beyond a LAB, then the effect of this variation will also have to be compensated.

Once the time between trigger event and the next DDS clock is calculated, the sampled trigger event starts the phase accumulator with a phase offset, which is equal to $t \times FCW \times Fclk$, where t is the measured time, FCW is the frequency control word and $Fclk$ is the DDS clock frequency.

The triggering of the DDS pulse generator could also be used to perform pulse period modulation (PPM) where the period of the pulse is modulated without affecting the pulse shape. When the DDS phase accumulator is triggered, the accumulator resets, the current pulse is stopped, and a new pulse is started. Therefore, if a trigger signal representing a modulated period is used for triggering, the resultant output would be a period modulated pulse waveform. In the pulse generator, the parameters A, B, C, D, E, and F are calculated assuming the current maximum deviated period, such that these values would be valid for any period for the current settings. To generate the trigger signal, a separate phase accumulator is implemented where the period or frequency is set to the center period. This phase accumulator is then frequency modulated with the scaled modulating waveform to give the desired period deviation. The most significant bit output of this phase accumulator is then used as a trigger event to trigger the main pulse generator. This trigger will suffer from the inherent DDS jitter [5]. However, the jitter is fully deterministic [5] and can be converted into equivalent phase offset for the trigger event that is added to the phase accumulator output for each respective trigger.

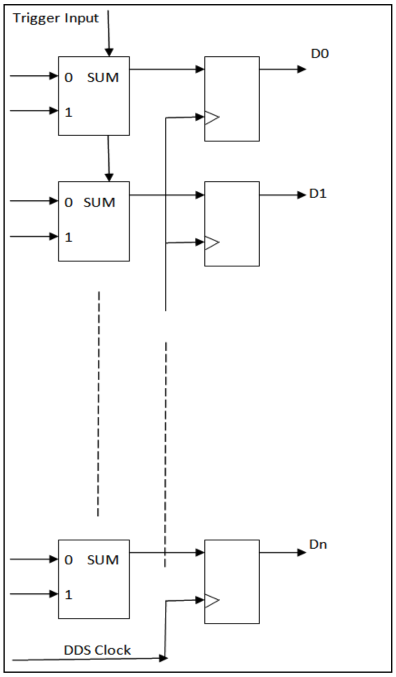


Figure 8. FPGA carry chain implemented as a delay line.

III. FPGA DESIGN SIMULATION AND SYNTHESIS

The paper achieved a complete prototype advanced pulse generator. Figure 9 is a block diagram representation of the waveform generator design. The design was complete with control section and analogue input and output hardware. The discussion in this section will be limited to the digital aspect of the design. Results presented in this section are screen shots of the actual outputs from the waveform generator measured in an oscilloscope. FPGA was chosen as the suitable platform to implement the digital design. The aim was to use a low-cost FPGA. FPGAs from different manufacturers and of different families were evaluated. An Intel Altera Cyclone IV FPGA was chosen for this design. Quartus design software was used for FPGA design. ModelSim was used for functional verification and debugging. Very high-speed integrated circuit hardware descriptive language (VHDL) was predominantly used to model the signal generator system. The mega function wizard of Quartus was also used for some specific designs.

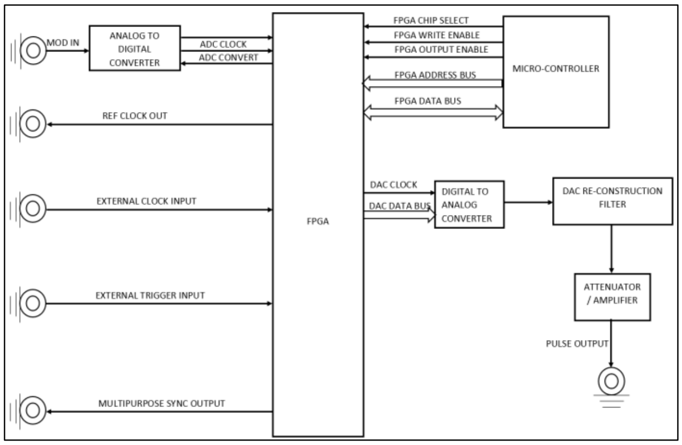


Figure 9. FPGA block diagram – input / output connections.

A. *FPGA Design*

The main phase accumulator block was implemented based on the design in the previous section. In addition to having an adder to add the scaled modulating waveform to the frequency control word to perform FM, another block was added to perform frequency shift keying (FSK). Two frequency control words were input to the FSK block which when enabled selects either of the inputs as the input to the phase accumulator based on the trigger level at every clock cycle. To perform triggering, a state machine was designed to generate a synchronous reset signal to the phase accumulator. Figure 10 presents the state flow diagram.

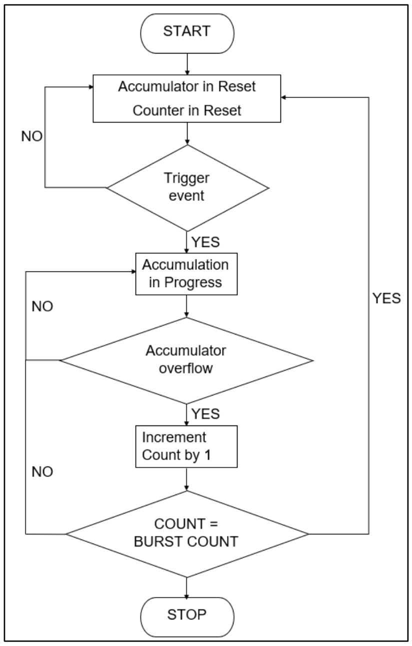


Figure 10. State flow diagram of the triggered phase accumulator.

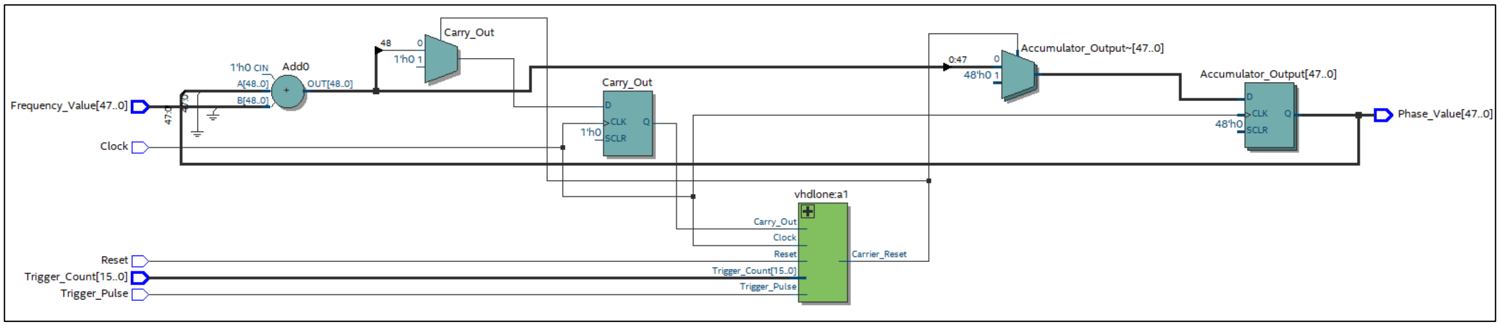


Figure 11. Simplified RTL view of the triggered phase accumulator.

When the trigger event happens, the synchronous reset is de-asserted. The accumulator starts running. The carry-out from the phase accumulator adder indicates the completion of a waveform cycle. A counter is incremented on every carry-out. When the count becomes equal to $N-1$, where N is the number of specified cycles, the synchronous reset is asserted, and the accumulator stops and then waits for the next trigger event to start a new burst. Figure 11 is a simplified RTL view of the triggered phase accumulator. The phase accumulator output is added with phase modulation output of the modulation generator, the phase of the carrier waveform and trigger uncertainty compensation phase factor is subtracted from the phase accumulator output at every clock cycle. A few pipeline stages are introduced to process the various additions and subtractions. The output of the phase adder is input to the pulse generator block.

The pulse generator is still essentially a DDS system and therefore suffers from jitter. However, having five samples on the pulse edges will guarantee that the edge time is accurately re-constructed by the filter following the DAC output, provided that the bandwidth of the filter is greater than the maximum pulse frequency and less than half the sampling rate of the clock frequency. Any remaining jitter on the edge appears not as timing jitter but as amplitude aberrations on the edges and could be further removed by the re-construction filter design. To achieve a minimum edge time of 5 nano-second and have five samples of the waveform on the pulse edges, the DDS clock frequency needs to be 800 MHz. It is not possible to run the core of a low-cost FPGA at this frequency. Therefore, four pulse generators, each running at 200 MHz is implemented and their outputs are multiplexed to produce samples at 800 mega samples per second before being sent to the DAC. For multiplexing, low voltage differential signal (LVDS), serializer de-serializer (SERDES) transmitter block in the FPGA is used, which could output data at 840 mega samples per second in the chosen FPGA. The transmitter block provides connection to the LVDS channels. This IP block is set appropriately to overcome any timing skew between the 200 MHz clock frequency, the 800 MHz clock frequency, the resultant high-speed serial data and the physical medium connection between the transmitter block and the DAC LVDS channels.

The external modulating signal is passed through an anti-alias filter before being digitised by the ADC. The sampling rate of the ADC is chosen to be much smaller than the DDS clock frequency to keep the cost of the ADC down. For a maximum modulation frequency of 100 kHz, ADC sampling rate of 12.5 MHz provides 125 samples for one cycle of the waveform which is sufficient for many complex waveforms.

Two ways of up converting the sampling rate of the external modulating signal to the DDS clock frequency were mentioned in the previous section. The CIC filter method was chosen because of its simplicity. The 12.5 MHz clock for the ADC is derived from the 200 MHz DDS clock using a clock divider. Therefore, the ADC data is synchronous to the DDS clock frequency. The input data is passed through a CIC filter to provide samples at every DDS clock frequency. A third order CIC filter with differential delay of one, up-converting the sampling rate from 12.5 MHz to 200 MHz for a pass-band frequency of 100 kHz would provide alias or image attenuation of more than 100 dB as analysed in [19]. It has also been described in [19] that for a CIC filter of order N, differential delay M, and rate change R, the number of output bits is equal to:

$$Bout = Bin + N \times \log_2 (R \times M) \tag{22}$$

where $Bout$ is the number of output bits and Bin is the number of input bits. For a third order filter with differential delay one and rate change equal to 16, the bit growth is 12 bits. For a 12-bit ADC this implies that all the additions and subtractions in the CIC filter should be carried out in 24 bits. The output of the CIC filter is truncated to 16 bits before being used as modulating waveform input. The number of bits in the CIC filter could have been reduced by pruning techniques as explained in [19] but this is not necessary as the chosen FPGA can easily perform 24 bits additions and subtractions at 200 MHz. The CIC filter does not have a wide flat passband. But for a 100 kHz pass-band at 12.5 MHz sampling rate, the droop in amplitude is less than 0.01 dB [19]. Therefore, there is no need for any compensation.

When external signal is used to trigger the generation of pulse waveform, the time between the start of the trigger and its registration on the next clock event is measured. The trigger signal is connected to the carry input of a logic element block in the FPGA. The carry out from the Logic Element (LE) is connected to the carry in of the following logic element and this process is continued for several logic elements to form a delay chain. Each of the logic elements is configured to perform addition of ‘0’ and ‘1’ and the carry input. The result is registered. When the trigger is low, the register output is high. When the trigger is high, the output is low, and the carry propagates through. The number of low register outputs on the next clock cycle is a measure of the time between trigger arrival and the next clock edge.

The carry-in, carry-out propagation delay for the chosen FPGA was less than 60 pico-second. There are 16 logic elements in a LAB. The delay chain extends beyond many LABs. The propagation delay between logic elements within a LAB is different from propagation delay between LABs which was approximately 180 pico-second. This difference is compensated by having two delay chains [22]. The trigger input to the second delay chain is delayed slightly by passing through some combinational logic. This delay will make the trigger cross the LABS at different times in the two chains. This time difference is then used to measure and sub-divide LAB propagation delay in multiples of LE propagation delay.

The carry chain propagation delay of each LE varies with temperature and process variations. To compensate for this, the delay line is made long enough to allow two clock edges to happen. The clock period is divided by the number of low register outputs between the two clock edges to give a measure of the delay of each LE. For example, if the number of low register outputs between clock edges is N1 in the first delay chain and N2 in the second delay chain, then the delay of each LE delay is calculated as follows.

$$LE_{DELAY} = \frac{2}{(fclk \times (N1 + N2))} \tag{23}$$

This is measured continuously and therefore account for temperature variations. 256 logic elements in the delay chain were found to be enough to accommodate two clock edges for a clock frequency of 200 MHz. Instead of relying on the FPGA fitter tool to place the logic elements in the FPGA, the logic elements are placed manually next to each other in the FPGA chip planner as shown in Figure 12. This is done to make the behavior of the delay line predictable and repeatable in a production environment.

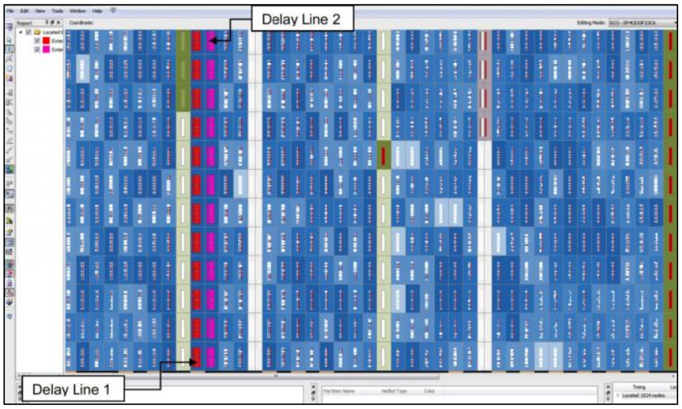


Figure 12. Trigger delay line placement in the FPGA.

Then looking at any one delay line, when the trigger goes high on any clock edge, if the number of low register outputs is equal to N, then the trigger to clock uncertainty is calculated as follows.

$$TRIGGER_{UNCERTAINTY} = LE_{DELAY} \times N \tag{24}$$

Once the trigger uncertainty is measured, it is multiplied by the frequency control word to convert it into a phase value. This is then subtracted from the output of the phase accumulator to provide uncertainty compensation of the trigger. The process of the calculation and compensation of trigger uncertainty introduces trigger latency, the time when the trigger arrives to the time when the trigger is used. This latency is several clock periods long. Trigger latency is not known to be a problem if it is fixed. The method described above reduces the trigger to output uncertainty from 5 nano-second to less than 60 pico-second. Measurement details are provided in the results section.

A microcontroller communicates with the FPGA to set various control words. The controlling processor is memory mapped to the FPGA. The communication does not have to be fast. These control words do not have to be changed on the fly. The selected microcontroller had built in external memory interface which was ideal to communicate with the FPGA. Figure 13 shows the connections between the microcontroller and the FPGA. The FPGA had its own dedicated chip enable from the microcontroller for exclusive communication.

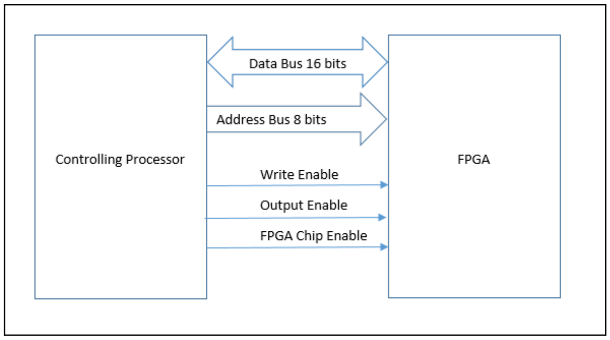


Figure 13. FPGA microprocessor interface.

When chip enable and write enable is low, the processor writes the 16-bit data on the 16-bit bi-directional data bus at the address specified by the 8-bit address bus. When chip enable and output enable is low, the processor reads the 16-bit data on the 16-bit bi-directional data bus from the address specified by the 8-bit address bus. The 8-bit address bus allows reading and writing of 256 16-bit control words which is more than sufficient for this application. The processor reads and writes asynchronously to the FPGA. However, the read and write cycles are slow. Therefore, it is not difficult to synchronize the control words to the main generator clock. If any control word is more than 16 bits long, there is a potential of setting incorrect control word momentarily while the processor is updating the control word. This is avoided by having two sets of registers in the FPGA, buffer registers and active registers. The processor updates the buffer registers, and the active registers are updated when all buffer registers have been updated.

B. FPGA Synthesis

The previous section charts the design considerations and implementation of the pulse waveform generator design. Once the design was verified, it was synthesized using Quartus II in a Cyclone IV E FPGA, EP4CE55F23C6. The summary of synthesis is presented in Table 2. The summary is for a complete pulse, function / arbitrary and noise generator. The pulse generator section of the design only uses memory for its modulation section. The carrier section does not use any memory. Following synthesis, the design was fitted in the FPGA. The design was analysed for timing requirements. Finally, programming files were generated to allow programming or configuration of the device. All these procedures were performed by the Quartus II software tool provided by Intel Altera.

Table 2. Quartus flow summary for the pulse generator project.

Flow Status	Results
Quartus II Version	13.1 Build 163 10/23/2013 SJ Web Edition
Revision Name	Pulse_Generator_TopLevel
Top-level Entity Name	Pulse_Generator_TopLevel
Family	Cyclone IV E
Device	EP4CE55F23C6
Timing Models	Final
Met timing requirements	Yes
Total logic elements	36,011 / 55,856 (64 %)
Total combinational functions	20,519 / 55,856 (37 %)
Dedicated logic registers	29,857 / 55,856 (53 %)
Total registers	29959
Total pins	181 / 325 (56 %)
Total virtual pins	0
Total memory bits	1,286,144 / 2,396,160 (54 %)
Embedded Multiplier 9-bit elements	184 / 308 (60 %)
Total PLLs	2 / 4 (50 %)

The design was also analysed for power consumption using 'PowerPlay Power Analyzer Tool' in Quartus II. The results were used for power supply designs for the FPGA. A schematic was prepared for the FPGA. Input and output signals were assigned to various pins in the FPGA. Synthesis was carried out again to check the validity of the assignments. Schematic was prepared to allow for the configuration of the FPGA in joint test action group (JTAG) programming mode during development and in passive serial (PS) programming mode by the micro controller in production. The schematic was prepared using configuration application note provided by Intel Altera. Finally, FPGA interface to micro controller and other analogue hardware were finalised.

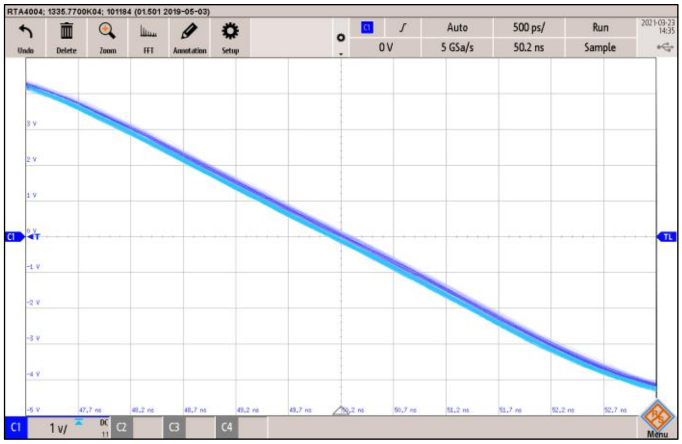
After preparing the FPGA schematic for the instrument, the microcontroller-based control section and the analogue hardware section were integrated with the FPGA section to prepare the complete generator schematic. The printed circuit board (PCB) layout was then prepared, and the prototype was built, followed by the testing and validation of the functional prototype.

C. Results

The output waveforms from the prototype board were analysed. Results are presented in this sub-section. Figure 14 shows screenshots that are measured outputs from a Tektronix MSO2012 oscilloscope where the waveforms are generated by the prototype board. The figure shows pulse waveforms with various delays, periods, widths, rise and fall times. All parametric changes are glitch free and phase continuous. The figure also shows pulse waveforms with various modulations and a triggered pulse waveform with pulse count set to three.

The generator achieved a wide frequency range from 1 μHz to 66.66 MHz with 1 μHz resolution. The resolution of 1 μHz is guaranteed by design by having 112 bits in the DDS phase accumulator [5]. Pulse timing parameters could be set with 10 pico-second resolution. This was verified using Rohde & Schwarz RTA4004 oscilloscope and statistically using Fluke PM6690 frequency timer / counter / analyzer. In Figure 15, two screenshots from the RTA4004 oscilloscope are laid on top of each other where the pulse width in one of the screenshots is incremented by 100 pico-second. An increment of 100 pico-second is chosen to show this figure as it is visually not possible to see 10 pico-second increments with a 500 pico-second time base which was the minimum time base that could be set in the RTA4004 oscilloscope.

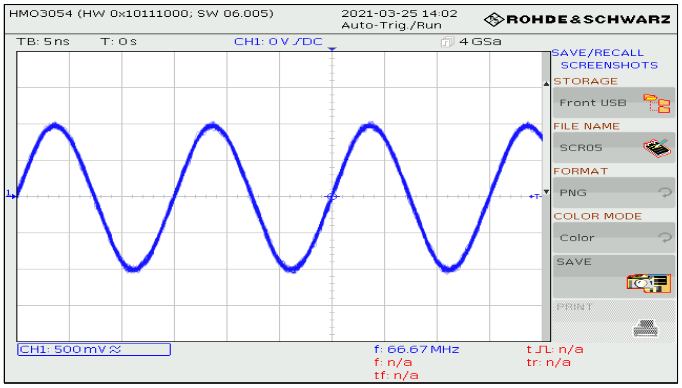
Figure 16 is a screenshot from Rohde & Schwarz HMO3004 oscilloscope which shows the generator output with period set to 15 nano-second, width set to 7.5 nano-second and edge time set to 5 nano-second. Figure 17 is the measurement results from Tektronix series 5 MSO oscilloscope with built-in jitter analysis software package. Figure 17 (a) shows that the pulse waveform root mean square (RMS) jitter accuracy is better than 30 pico-second. Figure 17 (b) shows that when the pulse is triggered externally with an asynchronous trigger pulse, the trigger to pulse output RMS jitter is better than 60 pico-second.


RTA4004; 1335.7700K04; 101184 (01.501 2019-05-03)

C1 ∫ Auto 0 V 5 GSa/s 500 ps/ 50.2 ns Run Sample 2021-03-23 14:35

C1 1 V/ DC 11

Figure 15. Scope measurement – pulse width increment.


HMO3054 (HW 0x10111000; SW 06.005) 2021-03-25 14:02 Auto-Trig./Run

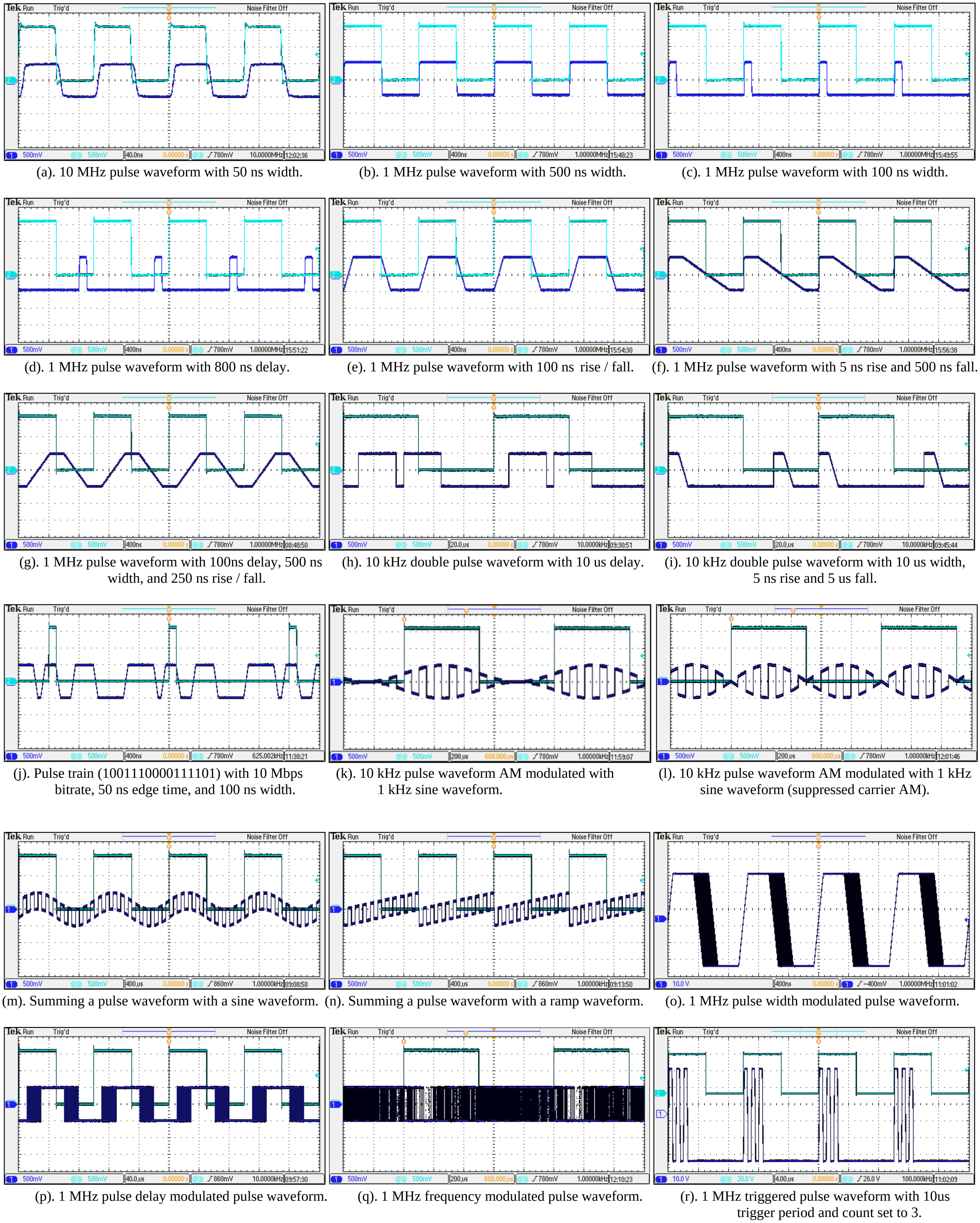
TB: 5 ns T: 0 s CH1: 0 V ∫DC 4 GSa

CH1: 500 mV ≈

f: 66.67 MHz
f: n/a
tf: n/a
t ⎍: n/a
tr: n/a

STORAGE: Front USB
FILE NAME: SCR05
FORMAT: PNG
COLOR MODE: Color

Figure 16. Scope measurement – minimum period, width, and edge times.



(a). 10 MHz pulse waveform with 50 ns width.

(b). 1 MHz pulse waveform with 500 ns width.

(c). 1 MHz pulse waveform with 100 ns width.

(d). 1 MHz pulse waveform with 800 ns delay.

(e). 1 MHz pulse waveform with 100 ns rise / fall.

(f). 1 MHz pulse waveform with 5 ns rise and 500 ns fall.

(g). 1 MHz pulse waveform with 100ns delay, 500 ns width, and 250 ns rise / fall.

(h). 10 kHz double pulse waveform with 10 us delay.

(i). 10 kHz double pulse waveform with 10 us width, 5 ns rise and 5 us fall.

(j). Pulse train (1001110000111101) with 10 Mbps bitrate, 50 ns edge time, and 100 ns width.

(k). 10 kHz pulse waveform AM modulated with 1 kHz sine waveform.

(l). 10 kHz pulse waveform AM modulated with 1 kHz sine waveform (suppressed carrier AM).

(m). Summing a pulse waveform with a sine waveform.

(n). Summing a pulse waveform with a ramp waveform.

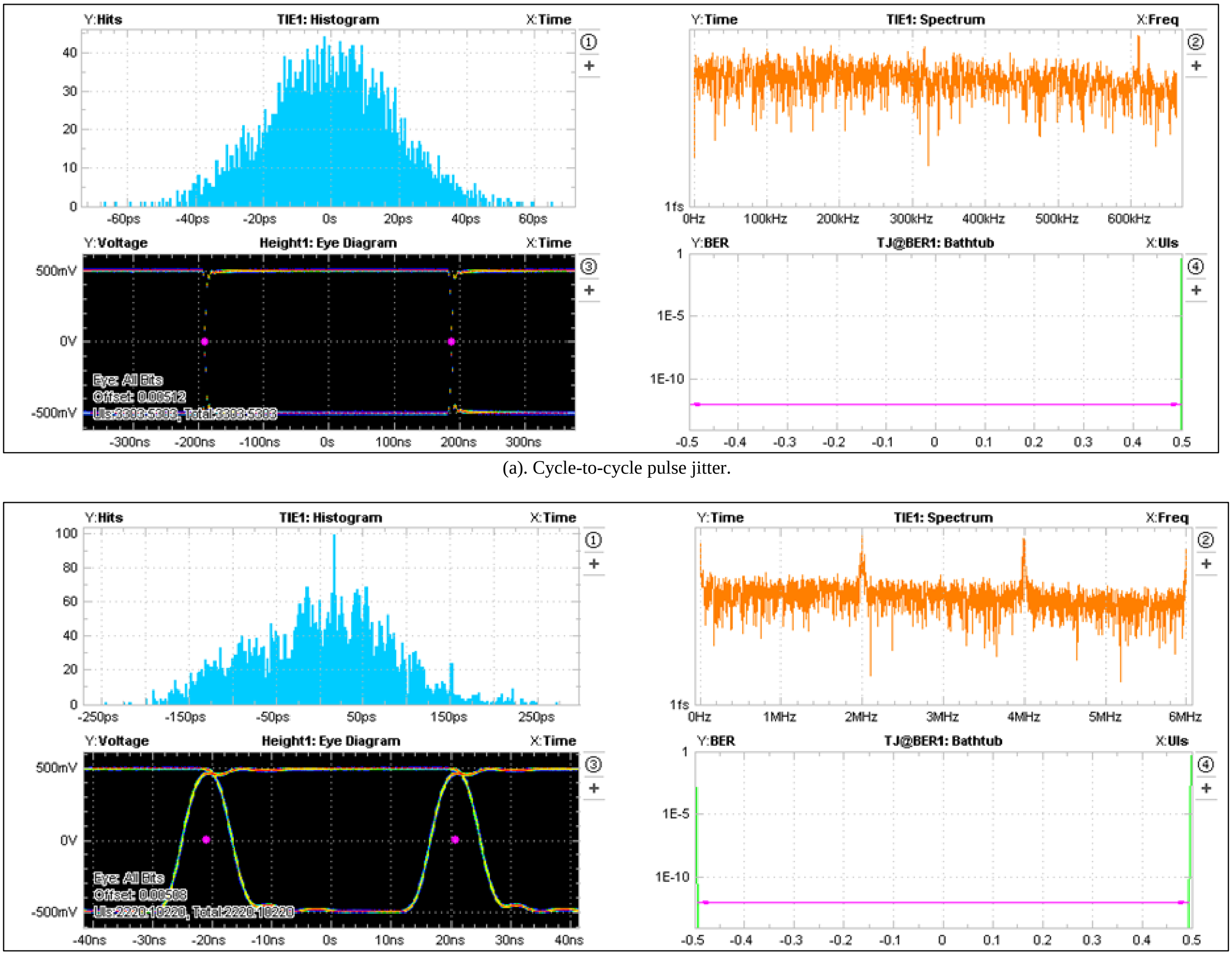
(o). 1 MHz pulse width modulated pulse waveform.

(p). 1 MHz pulse delay modulated pulse waveform.

(q). 1 MHz frequency modulated pulse waveform.

(r). 1 MHz triggered pulse waveform with 10us trigger period and count set to 3.

Figure 14. Scope measurement – Pulse waveforms including modulated pulses.



(a). Cycle-to-cycle pulse jitter.

(b). Trigger to output pulse jitter.

Figure 17. Jitter measurements.

Table 3 shows the improvements offered by the novel method over pulse generators based on conventional DDS architecture such as [7]. In comparison to other pulse generator architectures, the proposed design offers resolution of less than 10 pico-second in the setting of all pulse parameters including rise and fall times which is not possible in a pulse output DDS design such as [23] and still offers less than 30 pico-second RMS jitter. It offers considerable improvement in delay and width resolutions over counter-based pulse generator design such as [24] and [25] where width resolutions are limited to the clock period which is typically a few nanoseconds. Finally, in comparison to an analog pulse generator design [26], the period setting accuracy is only affected by the clock accuracy, typically 0.0005 %. Period accuracy is typically 5 % and 0.01 % in a variable frequency oscillator (VFO) and phase locked loop (PLL) based pulse generator designs, respectively. In an analog design and designs with pulse edge shaping circuit in the analog domain, rise and fall times can only be set up to a maximum of typically hundred milliseconds with limits for maximum rise and fall time ratio. Modulation is also difficult to achieve in an analog design. The proposed design overcome all these limitations.

IV. CONCLUSIONS

This paper has demonstrated the implementation of an entirely digital and modular, system on chip design of a versatile pulse waveform generator in a generic low-cost FPGA. A detailed overview of the principles behind the design, as well as simulation, functional verification and implementation of the concepts has been provided. The design of the pulse generator is completely modular and could easily be extended to output a double pulse or quad pulse waveforms with independent width, delay and rise and fall times setting for each pulse. The design also incorporated comprehensive modulation capabilities including AM, FM, PM, pulse width modulation, pulse delay modulation, pulse period modulation and sum modulation. Other modulation schemes such as

Table 3. Improvements offered by the new design over DDS.

	DDS			Proposed design		
Parametric changes	Re-loading of the waveform memory is required which will cause glitches and phase discontinuity at the pulse output.			All parametric changes are glitch free and phase continuous.		
Resolution	Resolution of pulse delay, width, and edge times are dependent on the memory depth and period setting.			Resolution of all pulse parameters are fixed and independent of other pulse parameters. Resolution of less than 10 pico-second was achieved in this design.		
	Memory	Period	Resolution	Memory	Period	Resolution
	65536	1 s	> 15 µs	N/A	≤ 2800 s	< 10 ps
Minimum pulse width	Dependent on memory depth and pulse period setting.			Independent of pulse period. Pulse width of 7.5 nano-second for period ranging from 15 nano-second to 1000000 s was achieved in this design.		
	Memory	Period	Width	Memory	Period	Width
	65536	1 s	> 15 µs	N/A	N/A	≥ 7.5 ns
Memory	Waveform memory is required which requires huge power consumption			No memory is required to generate the carrier pulse waveform.		
Jitter	Generating triggered pulse waveform from external trigger suffers from an uncertainty of 1 clock period (typically few nanoseconds)			This was reduced to less than 60 pico-seconds by measuring the uncertainty and compensating for it.		
	Clock Frequency	Jitter		Jitter	≤ 60 ps (RMS)	
	200 MHz	5 ns				
	800 MHz	1.25 ns				
Modulation	Pulse period, width and delay modulations cannot be achieved			Pulse period, width and delay modulations are all achievable.		

quadrature phase shift keying, quadrature amplitude modulation and orthogonal frequency division multiplexing are also possible in a multi-channel pulse generator design. The design could be modified and targeted to a newer FPGA to achieve enhanced performance and it is also flexible to be migrated to an ASIC device.

V. References

[1] C.Yao, X.Zhang, F. Guo, S.Dong, Y.Mi and C. Sun, “FPGA-Controlled All-Solid-State Nanosecond Pulse Generator for Biological Applications”, *IEEE Transactions on Plasma Science*, vol. 40, no. 10, pp. 2366-2372, October 2012.

[2] X. Qin *et al.*, “An FPGA-Based Hardware Platform for the Control of Spin-Based Quantum Systems”, *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1127-1139, April 2020.

[3] J. Tierney, C. Rader, and B. Gold, "A digital frequency synthesizer," *IEEE Transactions on Audio and Electroacoustics*, vol. 19, no. 1, pp. 48-57, 1971.

[4] Y. Yu, W. Hong, Z. H. Jiang, H. Zhang, and C. Guo, "Multibeam Generation and Measurement of a DDS-Based Digital Beamforming Array Transmitter at Ka-Band," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3030-3039, 2019.

[5] J. Vankka and K. A. I. Halonen, *Direct Digital Synthesizers, Theory, Design and Applications* (The Springer International Series in Engineering and Computer Science). US: Springer US, 2001.

[6] M. D. Martino, D. D. Caro, D. Esposito, E. Napoli, N. Petra, and A. G. M. Strollo, "A Standard-Cell-Based All-Digital PWM Modulator With High Resolution and Spread- Spectrum Capability," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 11, pp. 3885-3896, 2018.

[7] W. F. Adad and R. J. Iuzzolino, "Arbitrary function generator using Direct Digital Synthesis," in *2012 Conference on Precision electromagnetic Measurements*, 2012, pp. 622-623.

[8] Y. H. Chen, T. H. Wang, S. C. Lin, J. H. Chen, and Y. J. E. Chen, "A Pulse-Modulated Polar Transmitter Using Direct Digital Synthesis for 5G NR Mobile Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 10, pp. 1894-1898, 2020.

[9] G. D’Amato, G. Avitabile, G. Coviello, and C. Talarico, "DDS-PLL Phase Shifter Architectures for Phased Arrays: Theory and Techniques," *IEEE Access*, vol. 7, pp. 19461-19470, 2019.

[10] L. A. Aranda, F. Garcia-Herrero, L. Esteban, A. Sánchez-Macián, and J. A. Maestro, "Radiation Hardened Digital Direct Synthesizer With CORDIC for Spaceborne Applications," *IEEE Access*, vol. 8, pp. 83167-83176, 2020.

[11] S. K. Sullivan, R. L. Veith, I. Akiyama, Y. Fujisawa, and Y. Aizawa, "Triggered DDS Pulse Generator Architecture”, Patent " USUS7281025 B2, October 9, 2007.

[12] J. Wu, Z. Shi, and I. Y. Wang, "Firmware-only implementation of time-to-digital converter (TDC) in field-programmable gate array (FPGA)," *2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No.03CH37515)*, Portland, OR, USA, 2003, pp. 177-181 Vol.1.

[13] A. Sharma, "Design and Implementation of a Re-Configurable Arbitrary Signal Generator and Radio Frequency Spectrum Analyser," PhD, Univeristy of Hertfordshire, Hatfield, United Kingdom, 2016.

[14] Y. S. Kim, J. Lee, Y. Hong, J. E. Kim, and K. Baek, "Low-power pipelined phase accumulator with

sequential clock gating for DDFSs," *Electronics Letters*, vol. 49, no. 23, pp. 1445-1446, 2013.

[15] B. C. Rutherford and C. S. Lewis, "Practical direct digital synthesis for realizing high frequency signals from low frequency domains," in *2018 IEEE Radar Conference (RadarConf18)*, 2018, pp. 0670-0675.

[16] N. Tai, H. Lin, Y. Lu, H. Han, X. Xu, and Y. Zeng, "A Two-Dimensional Frequency Modulation Deception Jamming Against ISAR," *IEEE Access*, vol. 7, pp. 92079-92090, 2019.

[17] J. W. Lewis, "Coding a 40x40 Pipelined Multiplier," presented at the The Military and Aerospace Programmable Logic Device (MAPLD) International Conferences, Laurel, Maryland, 2002.

[18] R. Uusikartano, J. Nittylahti, and M. Renfors, "Area-optimized FPGA implementation of a digital FM modulator," in *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1999, vol. 4, pp. 360-362 vol.4.

[19] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155-162, 1981.

[20] J. Kalisz, "Review of methods for time interval measurements with picosecond resolution," *Metrologia*, vol. 41, no. 1, pp. 17-32, 2003 2003.

[21] H. J. M. Veendrick, "The behavior of flip-flops used as synchronizers and prediction of their failure rate", *IEEE J. Solid-State Circuits*, vol. SC-15, no. 2, pp. 169-176, April 1980.

[22] Y. Song, H. Liang, L. Zhou, J. Du, J. Ma, and Z. Yue, "Large dynamic range high resolution digital delay generator based on FPGA," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 2116-2118.

[23] H. Omran, K. Sharaf and M. Ibrahim, "An all-digital direct digital synthesizer fully implemented on FPGA," *2009 4th International Design and Test Workshop (IDT)*, Riyadh, pp. 1-6, 2009.

[24] 33210A Waveform/Function Generator, details available at : https://www.keysight.com/gb/en/products/waveform-and-function-generators/33210a-waveform-and-function-generator.html [accessed 20/01/2021].

[25] J. Liu, C. Huang and P. Lee, "A High-Accuracy Programmable Pulse Generator With a 10-ps Timing Resolution," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 621-629, April 2018.

[26] 4033 and 4034 50 MHz Pulse Generators, details available at : https://www.bkprecision.com/products/signal-generators/4033-50-mhz-pulse-generator.html [accessed 20/01/2021].

Ashok Sharma received the B.Eng. degree in instrumentation and control engineering in 2003, from the University of Pune, India, M.Sc. degree in embedded digital systems in 2005, from the University of Sussex, U.K. and Ph.D. degree in communications and electronics engineering in 2016, from the University of Hertfordshire, U.K. His PhD thesis is concerned with the design and implementation of arbitrary signal generator and radio frequency spectrum analyser. He was a research associate at the University of Hertfordshire. He is currently a senior electronics design engineer at Thurlby Thandar Instruments Limited, Huntingdon, U.K. In his current and previous roles, he has been involved in the designing of several test and measurement instruments including high frequency function generators, arbitrary waveform generators, pulse generators and radio frequency generators and spectrum analysers.

Yichuang Sun (M’90-SM’99) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

Dr. Sun is currently Professor of Communications and Electronics, Head of Communications and Intelligent Systems Research Group, and Head of Electrical and Electronic Engineering Discipline in the School of Physics, Engineering and Computer Science of the University of Hertfordshire, UK. He has published over 350 papers and contributed 10 chapters in edited books. He has also published four text and research books: Continuous-Time Active Filter Design (CRC Press, USA, 1999), Design of High Frequency Integrated Analogue Filters (IEE Press, UK, 2002), Wireless Communication Circuits and Systems (IET Press, 2004), and Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits - the Systems on Chip Approach (IET Press, 2008). His research interests are in wireless and mobile communications, microelectronic circuits and systems, instruments and measurement, and machine learning and AI.

Professor Sun was Series Editor of IEE Circuits, Devices and

Systems Book Series (2003-2008). He was Associate Editor of IEEE Transactions on Circuits and Systems I: Regular Papers (2010-2011, 2016-2017, 2018-2019). He was sole or lead Guest Editor of eight IEEE and IEE/IET journal special issues: High-frequency Integrated Analogue Filters in IEE Proc. Circuits, Devices and Systems (2000), RF Circuits and Systems for Wireless Communications in IEE Proc. Circuits, Devices and Systems (2002), Analogue and Mixed-Signal Test for Systems on Chip in IEE Proc. Circuits, Devices and Systems (2004), MIMO Wireless and Mobile Communications in IEE Proc. Communications (2006), Advanced Signal Processing for Wireless and Mobile Communications in IET Signal Processing (2009), Cooperative Wireless and Mobile Communications in IET Communications (2013), Software-Defined Radio Transceivers and Circuits for 5G Wireless Communications in IEEE Transactions on Circuits and Systems-II (2016), and Cognitive and AI-enabled Wireless and Mobile Communications in IET Communications (2020). He has also been on various IEEE society technical committees and IEEE international conference program committees.

Oluyomi Simpson (M’15) received the B.Eng. (Hons) degree in electrical and electronic engineering in 2007, M.Sc. degree in radio and mobile communication systems in 2008 and Ph.D. degree in communications and electronics engineering in 2016, all from the University of Hertfordshire, Hatfield, U.K. His PhD thesis is concerned with cognitive and cooperative wireless communications. He is currently a Senior Lecturer in communications and electronics engineering, academic lead of the communications lab and a key researcher in the wireless communication and RF systems research lab with the School of Physics, Engineering and Computer Science at UH. His research interests and technical expertise are in wireless communication, RF systems and instruments and measurement including cognitive radio communication systems, cooperative and relay networks and communications, RFID technologies and applications, RF energy harvesting communications, machine learning in communications, physical layer security and indoor localization. His work has been published and presented in numerous journals and leading international conferences. He was Guest Editor of IET journal special issues: Cognitive and AI-enabled wireless and mobile communications. He has also been widely involved in various IEEE technical committee and international conference activities. He has been a reviewer of several IEEE and IET journals and international conferences.