Architecture of RAM-Accumulator based Multichannel Digital PWM

Norma Hermawan
Biomedical Engineering Department
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
norma.hermawan@bme.its.ac.id

***Abstract*—Traditional counter based digital Pulse Width Modulator (PWM) has been a straightforward architecture. However, realization of multichannel PWM with such design requires considerable number of logic gates. A novel architecture of multichannel PWM based on the usage of RAM block and frequency word accumulator is proposed. This architecture is capable of producing numerous channels while maintaining its performance. The design was tested in Microsemi SmartFusion A2F200 Customizable System on Chip, giving the result of 7.73% FPGA usage to produce 48 channels of 380Hz 16 bits PWM. The output performance for test application is acceptable.**

Keywords—PWM; RAM; accumulator; multichannel; digital

I. Introduction

Pulse width modulation (PWM) is a technique for controlling analog circuits with a digital output. It is employed in various applications such as lighting, measurement, communications and power conversion control.

An analog signal has a continuously varying value, with infinite resolution in both time and magnitude, whereas analog circuit is any device whose output is linearly proportional to its input. Despite of analog circuit control simplicity, it is not always practical. Analog signal is highly susceptible to any perturbation or noise as a consequence of its infinite resolution. Controlling analog circuits digitally reduces system costs and power consumption drastically. In addition, available microcontrollers and DSPs include on-chip PWM controllers to alleviate implementation.

In the PWM, an analog signal level is encoded into duty cycle of a square wave. The PWM signal is considered as digital signal since the output is in a certain state of either fully on or fully off. The voltage or current source is supplied to the analog load by means of a repetitive on and off pulses. During on phase, DC supply is applied to the load. On the contrary, supply is switched off during period of off phase. The ratio between on period and a full cycle PWM period is what so called duty cycle. The average power transfer then can be calculated considering the duty cycle of PWM voltage or current source [1].

In this paper, a digital architecture of multichannel PWM is proposed. The objective of the proposed system is to reduce the gate usage of traditional digital PWM design by moving register portion to RAM block.

II. Traditional Pulse Width Modulator (PWM)

A basic analog PWM as seen on Fig. 1 is built from an op-amp based comparator circuit and triangle generator, beside an analog input that may be in the form of a voltage level. The comparator works by constantly compare saw tooth with an analog input signal. The output remains high as long as the saw tooth voltage is less than the analog control signal. When that saw tooth exceeds the analog input, the comparator output turns low. As a result, raising the voltage level of analog input will increase the pulse width of the PWM signal [2].

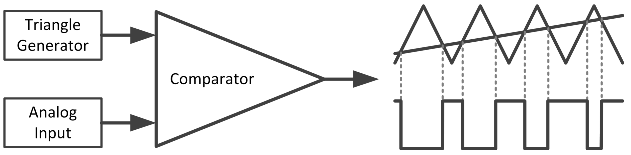


Fig. 1. Principle of a PWM Signal Generator

Both comparator and triangle wave generator is not necessarily op-amp based circuitry. They can be built from transistors or remarkable 555 timer IC as well. The 555 datasheet suggests application note for building a linear ramp circuit. Referring the same datasheet, Fig. 2 shows the 555 based PWM circuit. When the timer is connected in the monostable mode and triggered with a continuous pulse train, the output pulse width can be modulated by a signal applied to pin 5 [3].

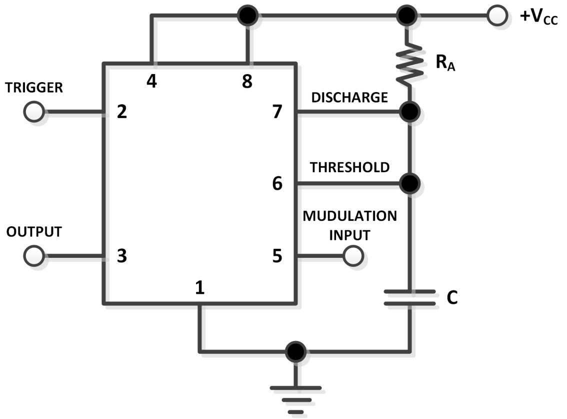


Fig. 2. 555 based PWM circuit

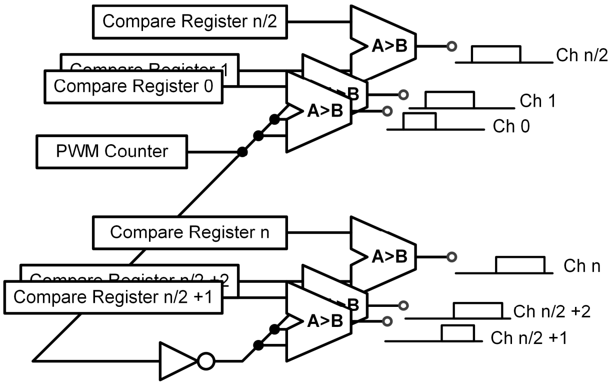


Fig. 3. Digital Multichannel PWM

Comparable to the analog circuit, a digital PWM is composed of a digital counter, a compare register and a digital comparator. Indeed, an array of comparators and compare registers are required when multi-channel design is desired. Fig. 3 shows the diagram block of multichannel digital PWM.

In accordance to the Fig. 4, the counter increments its value from zero to predefined maximum number 2^B. At any given instant of time, all comparators in the first half of PWM channels compare the counter value with their corresponding register value. If the counter value is lower than register value, the corresponding PWM channel produces high output. For the second half, the comparator inputs are fed with inverted value of the counter output to make the counter act as a down counter. As a result, the PWM signal is aligned to the upper bound of counting cycles. This alignment is attempted to distribute power peak to be approximately equal whole time.

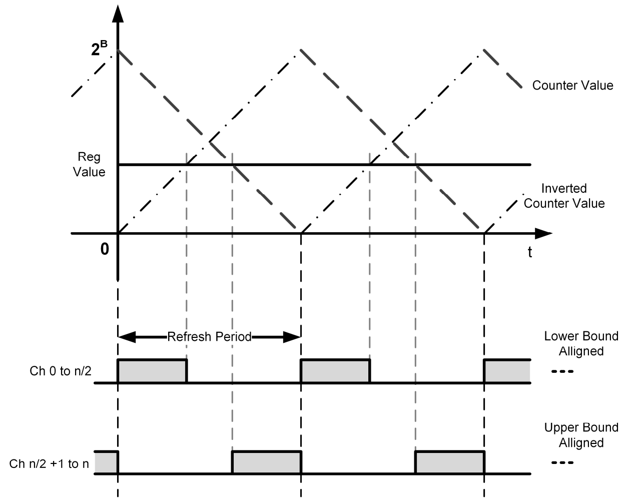


Fig. 4. Counter Value and PWM Alignment

The refresh rate of PWM relies upon clock frequency and the PWM resolution, which is given by

$$f_{PWM} = \frac{f_{clk}}{2^B} \tag{1}$$

Where
f_{PWM} = PWM refresh rate
f_{clk} = PWM clock rate, and
B = PWM resolution

An accumulator has been proposed as replacement of counter in the architecture of traditional PWM [4]. As shown in Fig. 5, a frequency word input is continuously accumulated into register. As a result, the output of this block produces a saw tooth which slope is digitally controllable by the frequency word input. The refresh rate of this PWM thus generated by the frequency word, instead of counter width. One of advantage of employing accumulator is the possibility of generating variable phase PWM, which is known as Phase-Accumulator PWM [5]. Further exploration discovers the implementation of accumulator for optimized design of multichannel PWM in devices with RAM blocks, as presented in this paper.

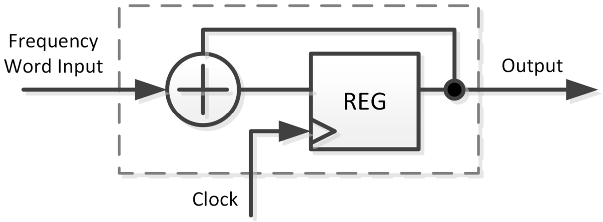


Fig. 5. Accumulator as Counter Replacement

III. Design Architecture

Programmable devices have been far developed since it was first introduced. They are in the form of FPGA and customizable system on chips. Most FPGAs have embedded memory structure that is separated from main logic elements. Design architecture in this paper is optimized for such devices with embedded RAM blocks.

A. RAM Based PWM

Traditional counter based multichannel PWM requires a number of register array as a compare value for each channel. The purpose of developing RAM based PWM is to transfer these registers allocation into RAM blocks which have their own allocation aside from valuable portion of FPGA or customizable system on chip, the logic block. This idea is exceptionally brilliant to save gate usage for certain logic devices. The detailed principle of RAM Based PWM generator is illustrated in Fig. 6.

An up counter in this architecture is divided into two fields, the lower N bits is Address counter for counting up RAM address, the next B bits is PWM main counter. The PWM main counter increments its value from 0 to its maximum value, i.e. 2^B. At each step of increment, the value of PWM counter is compared to all of RAM contents. Since RAM content is only accessible one address at any given instant of time, a single PWM value comparison must be multiplexed by number of PWM channel 2^N. As a consequence, the PWM refresh rate is divided by 2^N. In the other word, the frequency of RAM access must be a 2^N multiple of PWM rate times resolution.

$$f_{RAM} = 2^N 2^B f_{PWM} \tag{2}$$

Thus, if a single clock cycle represents a period of RAM access, the period of a PWM cycle is

$$T_{PWM} = 2^N 2^B T_{clk} \tag{3}$$

Hence

$$f_{PWM} = \frac{f_{clk}}{2^N 2^B} \quad (4)$$

Where f_{RAM} is maximum access frequency of RAM, f_{PWM} is PWM rate, B is PWM resolution and 2^N is number of PWM Channel.

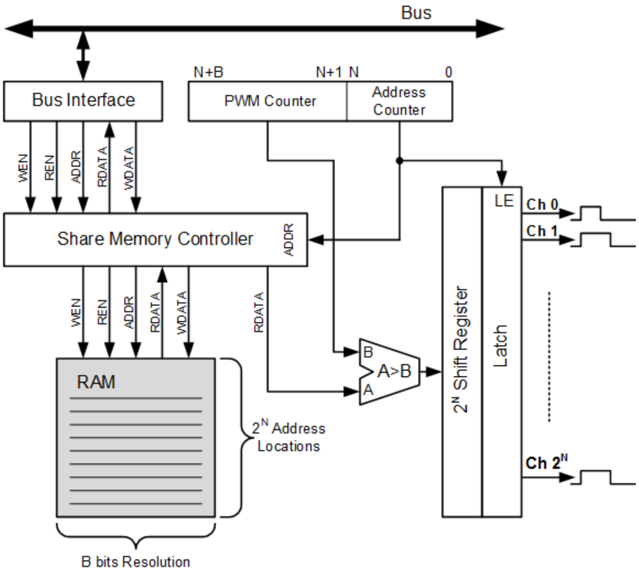


Fig. 6. RAM Based PWM Signal Generator

In practice, the number of PWM channel is arbitrary and must not be bounded in a power factor of 2^N. Thus, the counter is modified so that the maximum value of Address Counter is limited to any value lower than 2^N before the PWM Counter start incrementing its value. In this case, the frequency of PWM is

$$f_{PWM} = \frac{f_{clk}}{n_{ch} 2^B} \quad (5)$$

Where n_{ch} in number of PWM channels. Fig. 7 shows the timing diagram of RAM based PWM for certain n_{ch} channels and B resolution.

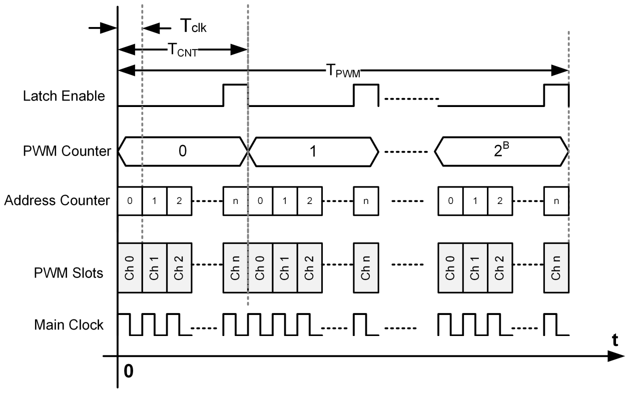


Fig. 7. Timing Diagram of RAM based PWM Generator

Parallelizing strategy is applicable to obtain more PWM channels if extra RAM blocks are available. Fig. 8 shows the utilization of multiple RAM in parallel configuration for that reason. The memory controller is duplicated to manage dataflow from RAM to comparators and bus interface. The address counter and main PWM counter may be shared for all modules to save more logic gates. Hence, for the specified PWM rate f_{PWM}, PWM resolution B and a number of RAM block n_{RAM}, the obtainable number of PWM channel is

$$n_{ch} = \frac{f_{clk}}{f_{PWM} 2^B} n_{RAM} \quad (6)$$

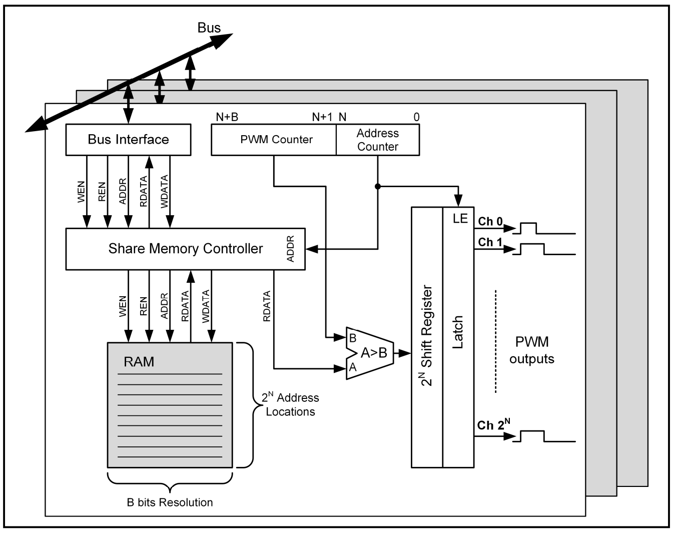


Fig. 8. Parallel RAM Architecture

B. Combining RAM and Accumulator

The advantage of employing accumulator in PWM is its multiple frequency effect. This feature is applicable to overcome the drawback of previously explained RAM based PWM generator. As on the Eq. (4), the lower PWM rate caused by non-concurrent RAM access is compensable to meet minimum required PWM refresh rate. The proposed system is illustrated in Fig. 9.

Fundamental frequency of single RAM based PWM is given by Eq. (5). Given the multiple effect of f^d, the resulting PWM rate can be calculated as

$$f_{PWM} = f^d * \Delta f \quad (7)$$

By substituting Eq. (5) into Eq. (7), the minimum multiplier frequency f^d is computable by applying the following formula

$$f_{PWM} = f^d \frac{f_{clk}}{n_{ch} 2^B} \quad (8)$$

hence,

$$f^d = \frac{n_{ch} 2^B f_{PWM}}{f_{clk}} \quad (9)$$

f^d is chosen from any prime number beyond the calculated value.

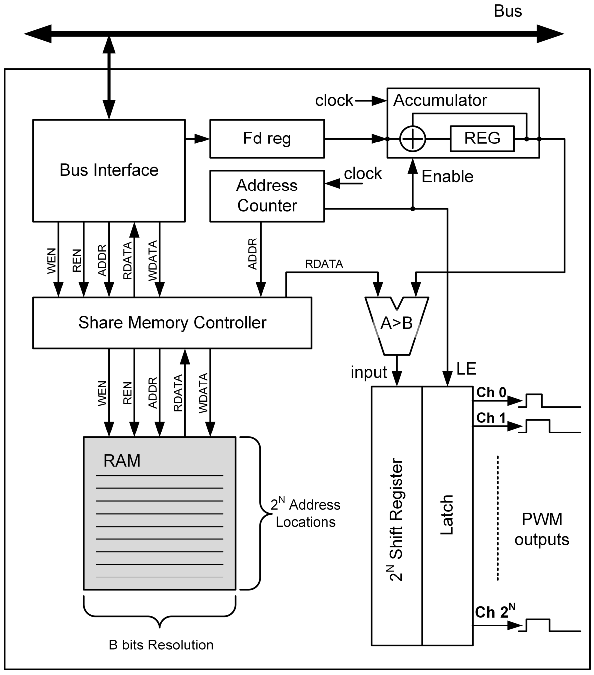


Fig. 9. RAM-Accumulator PWM Architecture

C. Channel Expansion

By combining RAM and accumulator technique, the maximum channel of PWM generator is entirely limited by the address size and the number of available RAM block on the device. The lower PWM rate as a result of multiplexed RAM access can be counterbalanced by the multiplier factor f^d. However, another critical issue that must be considered in practical design of PWM generator is regarding with number of device pins. Each package type on any device has different number of I/Os. When the desired channel is more than available pin, an expansion technique must be undertaken.

The expansion technique can be done by simply moving the shift register and latch component out from the device and replace it by an external logic circuit. However, the clock rate specification of this circuit must be equal or higher than the device clock. If clock specification of the circuit cannot catch up the device clock, the serial line can be split into several lower speed lines. The logic circuit can be built from either CMOS or TTL digital logic ICs.

Fig. 10 depicts the detailed principle of pins expansion method. The parallel outputs are serialized by some shift registers and external logic circuit is connected to revert it back into parallel form. The required pin for device is counted on the factor of registers width B_{reg}, the additional clock pin, latch enable pin and any additional special purpose pins as what is shown on Eq. (10).

$$n_{pin} = \frac{n_{ch}}{B_{reg}} + clk_{pin} + LE_{pin} + x_{pins} \qquad (10)$$

Hence, the maximum possible channel is

$$n_{ch} = (n_{pin} - clk_{pin} - LE_{pin} - x_{pins})B_{reg} \qquad (11)$$

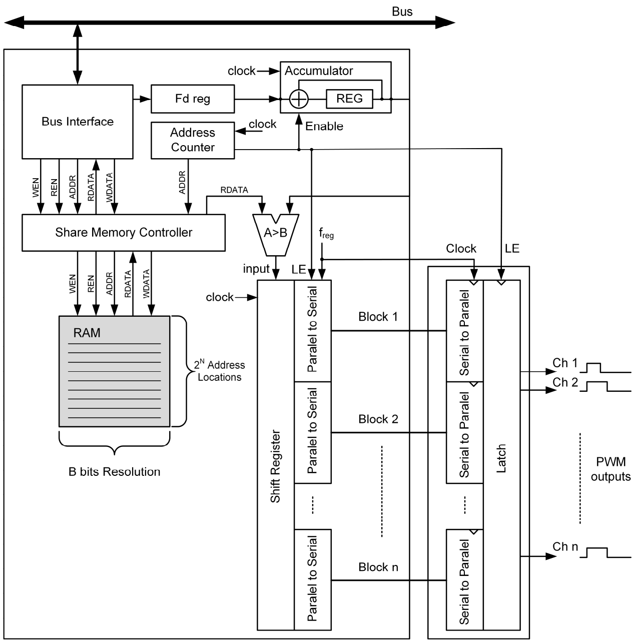


Fig. 10. Port Expansion Technique

where

n_{ch}	= number of channel
n_{pin}	= number of required FPGA I/Os
clk_{pin}	= 1
LE_{pin}	= 1
x_{pins}	= special purpose pins, and
B_{reg}	= external serial to parallel register

The required external circuit is equal to the number of multiplexed channels, that is

$$n_{block} = \frac{n_{ch}}{B_{reg}} \qquad (12)$$

The external logic circuit must run at least in frequency of register f_{reg} that is given by

$$f_{reg} = \frac{f_{clk}}{n_{block}} \qquad (13)$$

At this point, the constraint for generating multichannel PWM can be identified from maximum n_{block} which must satisfy both

$$n_{block} = n_{pin} - clk_{pin} - LE_{pin} - x_{pins} \qquad (14)$$

And

$$n_{block} = \frac{f_{clk}}{f_{reg}} \qquad (15)$$

Those mean the lower external circuit frequency, the more external circuit blocks and device pins are required, vice versa. The register width B_{reg} may be picked accordingly based on the obtained n_{block} and the number PWM channel n_{ch}. As a result, the boundary issue of generating multichannel PWM have shifted from the factor of pin number to the maximum clock of the glue logic.

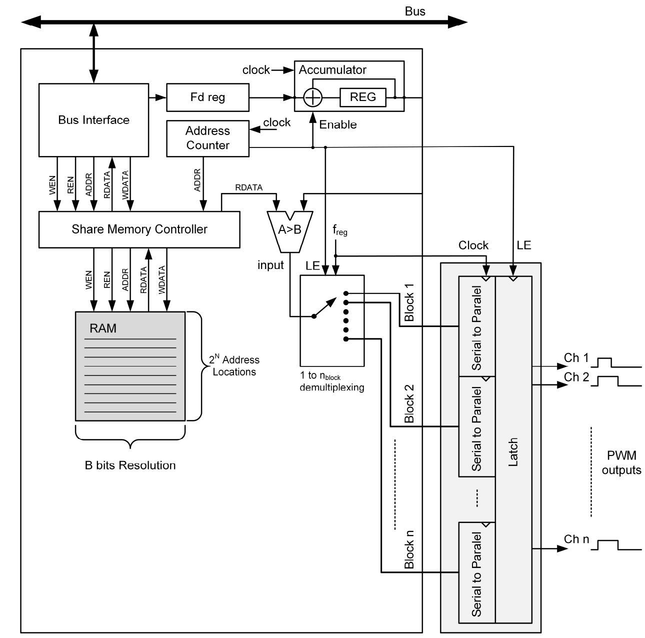


Fig. 11. Replacing Shift Register with Multiplexer

Saving PWM value in RAM is essentially the same mechanism as multiplexing PWM signals in a serial line. Shift registers are used to demultiplex this signal into parallel configuration. Port expansion technique has been introduced to overcome the number of available pin by multiplexing back the parallel signal into several serial lines. However, this whirl process can be simplified by replacing the shift registers in Fig. 10 with a multiplexer and latches. As a result, the redrawn design in Fig. 11 is much more efficient than previous approach.

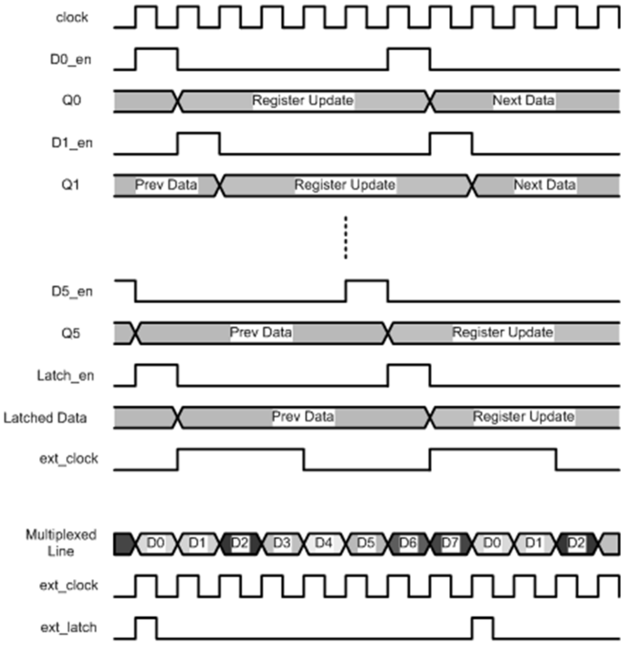


Fig. 12. Port Multiplexer Timing Diagram

The multiplexer circuit is implemented by referring timing diagram in Fig. 12 which is derived from the functionality of basic form. Instead of a combinational multiplexer circuit, it may be a form of shift register followed by the other flip-flop latches configuration. The hardware implementation of port multiplexer in Fig. 13 suggests that a lot of flip-flop reduction have been done.

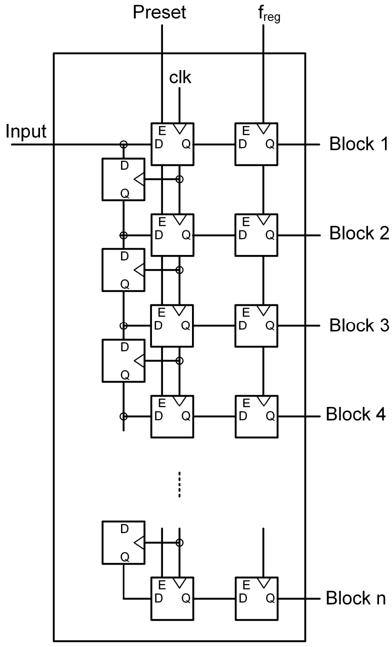


Fig. 13. Port Multiplexer Architecture

As a complement of port multiplexer, additional external hardware is required to parallelize and latch the signal. Critical constraints for choosing logic components for this external circuit is minimum speed specified by Eq. (13). An example that is already verified in this experiment is 74LV595 which functional diagram is shown in Fig. 14 [6].

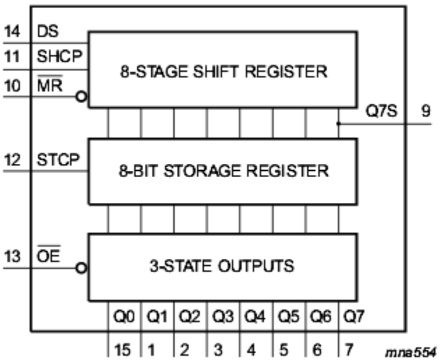


Fig. 14. Functional Diagram of 74LV595

IV. Hardware Implementation and Evaluation

The latest development of programmable digital device is combining the logic blocks of digital architectures and interconnects traditional FPGAs with embedded microprocessors and related peripherals to form a complete configurable system on chip. Such technologies are available in the Xilinx Zynq-7000 All Programmable SoC, Atmel FPSLIC, and the Microsemi SmartFusion Customizable System on Chip (cSoC) devices. The last device integrates an ARM Cortex-M3 processor and analog peripherals to their flash-based FPGA fabric with embedded SRAM Blocks [7]. The introduction of embedded microprocessors in a single chip indeed reduce the portion of FPGA logic gates in the relatively similar device size.

Implementation of RAM based PWM architecture in this paper has been tested on the Microsemi SmartFusion A2F200 cSoC device. The design requirement is to fit 48 channels of 300 Hz 16 bits PWM into the smaller SmartFusion family, A2F060.

A. Microsemi SmartFussion

The SmartFusion Customizable System-on-Chip (cSoC) is a device that integrates an FPGA, ARM® Cortex™-M3, and programmable analog blocks. SmartFusion cSoCs are designed for a true system-on-chip (SoC) solution that compromise flexibility and lower cost of hard processor cores. The SmartFusion family comes with 3 devices, A2F060, A2F200 and A2F500. The main building blocks of these device are classified in three feature groups, Microcontroller Subsystem (MSS), FPGA and Programmable Analog Front-End (AFE) and engine.

SmartFusion has microprocessor system block called Microcontroller Subsystem (MSS). MSS consists of an ARM Cortex-M3 processor and some complement peripherals such as Timer, UART, ethernet MAC, etc. The processor, peripherals, FPGA fabric and analog block are interconnected with the ARM Advanced Microcontroller Bus Architecture (AMBA) bus.

The SmartFusion FPGA fabric consists of abundant logic elements that called VersaTiles. A logic VersaTile cell has four inputs and one output. Each VersaTile can be configured using the appropriate flash switch connections to be one of various logics. Depends on SmartFusion cSOC family, the number of tiles are different. A SmartFusion tile is approximately equal to 40 system gates.

B. Design Testing

PWM itself is a simple digital architecture. However, fitting 48 channels of 16 bits PWM generator into 1,536 pieces of available logic elements of A2F060 expense significant effort as previously explained. The designs was tested on the SmartFussion A2F200 which has three times more logic elements than A2F060.

Microsemi CSoC products come with pre-implemented, synthesizable Intellectual Property (IP) building blocks which are designed, optimized and verified in the FPGAs with comprehensive documentation. One of those IPs is configurable PWM core which is available in the repository and ready to download anytime when it is needed. The firmware's are provided as well. Instantiation of IP cores is as simple as drag and drop the modules in the provided IDE tool, the Libero SoC.

Several configurations of Microsemi PWM core are tested as a benchmark for the later designs. Each design may instantiate many cores, and each PWM core can be configured for multiple channels. The summary of synthesis report for these configurations on SmartFusion A2F200 is presented in TABLE I. It can be overviewed that the maximum channels of 16 bits PWMs that feasible by deploying Microsemi IP cores to SmartFusion A2F200 is around 40.

The fact that PWM Core is vastly configurable was suspected to be inefficient. Some features such as tachometer that appears in the core configuration editor is never used and indeed could be removed.

TABLE I. SYNTHESIS SUMMARY FOR A2F200

Configuration				**Synthesis Result**
PWM Resolution	***Number of cores***	***Channels per core***	***Total Channel***	
32 bits	8	16	128	Error: 502.95% of Fabric modules are required
32 bits	3	16	48	Error: 205.64% of Fabric modules are required
16 bits	3	16	48	Error: 104.82% of Fabric modules are required
16 bits	5	8	40	Succeed. 91.38% of Fabric modules are required

However, by analyzing deeper through the source code from configurator, it is concluded that further optimization is impossible. The VHDL source code uses generate statements to enable or disable its features. As a result, parameterization from configuration editor does not alter efficiency of the design. In the other hand, removing any unused input-output port does not give notable effect as well.

```
=================
Compile report:
===============

    Microcontroller Subsystem  Used:      1  Total:
1   (100.00%)
    Fabric                     Used:   3139  Total:
4608   (68.12%)
    Fabric IO (W/ clocks)      Used:     48  Total:
94   (51.06%)
    Fabric Differential IO     Used:      0  Total:
47   (0.00%)
    Dedicated Analog IO        Used:      0  Total:
32   (0.00%)
    Dedicated MSS IO           Used:      9  Total:
43   (20.93%)
    GLOBAL (Chip+Quadrant)     Used:      3  Total:
15   (20.00%)
    MSS GLOBAL                 Used:      3  Total:
3   (100.00%)
    On-chip RC oscillator      Used:      1  Total:
1   (100.00%)
    Main Crystal oscillator    Used:      0  Total:
1   (0.00%)
    32 KHz Crystal oscillator  Used:      0  Total:
1   (0.00%)
    RAM/FIFO                   Used:      3  Total:
8   (37.50%)
    User JTAG                  Used:      0  Total:
1   (0.00%)
```

Fig. 15. Synthesis Result of Traditional Custom PWM Core

Approaching multichannel PWM with custom core gives slightly better outcome than by the stock IP Core. The custom 48 channels PWM use the same design as Fig. 3 that consists of 48x16 bits registers, 48 comparators, a counter and APB interface. Implementation of such design gives the synthesis report as in Fig. 15.

The SmartFusion A2F200 FPGA fabric has 8 SRAM blocks along the north side of the die which is a potential powerful feature to utilize. The maximum clock rate of SmartFusion A2F200 is 100 MHz. When this frequency is used to clock the 16 bits counter of PWM, the PWM refresh rate can be obtained from Eq. (1)

$$f_{PWM} = \frac{10^8}{2^{16}} \approx 1525\ Hz$$

If RAM based 48 channel PWM design as in Fig. 6 is used, comparator needs exactly one clock cycle to compare each data in the RAM with counter value. Hence, all channels finish refreshing their state after 48 clock cycle, and the refresh rate for all 48 channels PWM is

$$f_{PWM(0-47)} \approx \frac{1525}{48} \approx 31.8\ Hz$$

In order to increase this refresh rate, the design is parallelized to utilize multiple RAMs as in Fig. 8. Parallelizing the design is simply multiply the refresh rate by factor of RAM number. If all of 8 blocks RAM in SmartFusion A2F200 are used, the resulting refresh rate is

$$f_{PWM(0-47)} \approx 31.8 * 8 = 254.3\ Hz$$

Synthesis report of this design shows slightly better utilization of FPGA fabric compared with of the custom traditional design. The virtually large utilization is caused by memory controller duplication. Although this result is quite satisfying, the design is not fit with the smaller SmartFusion A2F060 (30% logic gates of A2F200).

The RAM-accumulator design as in Fig. 11 is tested with given specification, i.e.

f_{PWM}	= 300 Hz
f_{clk}	= 100 MHz
n_{ch}	= 48
B	= 16

From the Eq. (9), the f^d is obtained by

$$f^d = \frac{48 * 2^{16} * 300}{10^8}$$

$$f^d \approx 9.437184$$

Any prime number above f^d can be picked as multiplier frequency e.g. 11. When the value is substitute back to the formula, the resulting PWM refresh rate is

$$f_{PWM} = 11\frac{10^8}{48 * 2^{16}}$$

$$f_{PWM} \approx 349.69\ \text{Hz}$$

This value completely meets the design requirement.

The port expansion technique presented in Fig. 10 need to be tested as well. The external logic components use 20 MHz shift register ICs. According to Eq. (15), the required blocks is

$$n_{block} = \frac{10^8}{2 * 10^7}$$

$$n_{block} = 5$$

Therefore, the required number of pins follow Eq. (14), that is

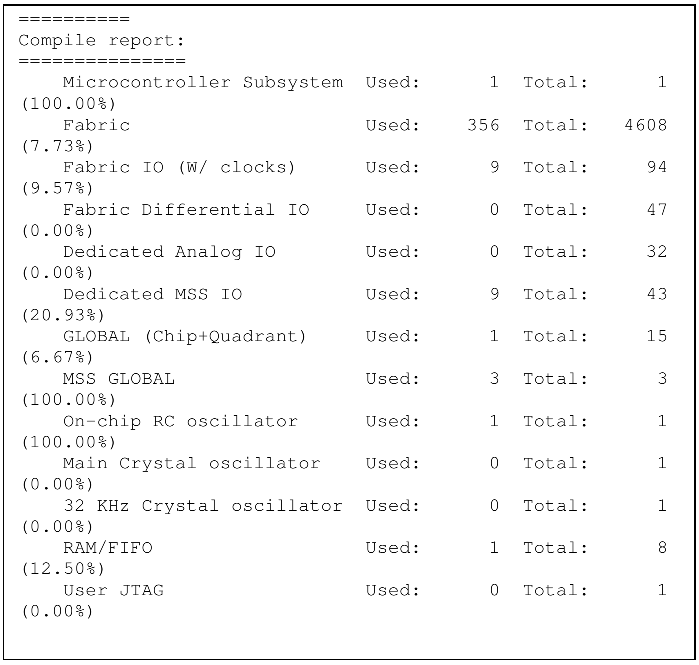
$$n_{pin} = 5 + 1 + 1 + x_{pins}$$

On the test application, a pair of I2C signals are required for controlling some variables resulting $x_{pins} = 2$. As a result,

$$n_{pin} = 5 + 1 + 1 + 2$$

$$n_{pin} = 9$$

This experiment demonstrates that any number of PWM channel can be generated by means of 9 FPGA pins only. The synthesis result for this configuration is given in Fig. 16.



```
==========
Compile report:
===============
    Microcontroller Subsystem  Used:      1  Total:      1
(100.00%)
    Fabric                     Used:    356  Total:   4608
(7.73%)
    Fabric IO (W/ clocks)      Used:      9  Total:     94
(9.57%)
    Fabric Differential IO     Used:      0  Total:     47
(0.00%)
    Dedicated Analog IO        Used:      0  Total:     32
(0.00%)
    Dedicated MSS IO           Used:      9  Total:     43
(20.93%)
    GLOBAL (Chip+Quadrant)     Used:      1  Total:     15
(6.67%)
    MSS GLOBAL                 Used:      3  Total:      3
(100.00%)
    On-chip RC oscillator      Used:      1  Total:      1
(100.00%)
    Main Crystal oscillator    Used:      0  Total:      1
(0.00%)
    32 KHz Crystal oscillator  Used:      0  Total:      1
(0.00%)
    RAM/FIFO                   Used:      1  Total:      8
(12.50%)
    User JTAG                  Used:      0  Total:      1
(0.00%)
```

Fig. 16. Synthesis Result of RAM-Accumulator based PWM

C. *Maximum Number of Channel*

Maximum number of channel that can be obtained with Microsemi SmartFusion is bounded by the number and size of RAM blocks, number of logic elements, and available FPGA IO. RAM size and number of block is specified in the datasheet of devices, while maximum logic control that can be generated from FPGA can be estimated by calculating synthesis report in Fig. 16. According to that result, A2F200 uses 7.73% logic element for each RAM block controller. Maximum of approximately 12 blocks can be controlled in that device.

Since A2F060 has 30% of logic blocks, it is capable to control approximately 4 blocks. Since both devices have the same number and size of RAM blocks, the maximum feasible channel is measurable. TABLE II. outlines these limitations.

TABLE II. DEVICE LIMITATION

Device	Approximated Limit		
	Logic Element Limit	***RAM limit***	***Total Feasible Channels***
A2F200	12 Blocks	8 of 512x18	4096 channels
A2F060	4 Blocks	8 of 512x18	2048 channels

In the other hand, the maximum channel is limited by availability of device IOs that is summarized in TABLE III. However, the expansion technique as explained in III.C can be employed to overcome the IO limitation by additional external shift register.

TABLE III. DEVICE I/O

Device	Package				
	TQ144	***CS288***	***PQ208***	***FG256***	***FG484***
A2F200	NA	78	66	66	94
A2F060	33	68	NA	66	NA

D. Size-Performance Trade-off

The main advantage of employing the presented design into FPGA or Customizable SoC is its flexibility feature. Due to availability of RAM block in most devices, a custom PWM design with abundant channels can be developed conforming to the requirements. The number of output channels can be raised by adjusting its performance. On the contrary, by limiting its channel number in certain devices, an extremely high performance output is obtainable.

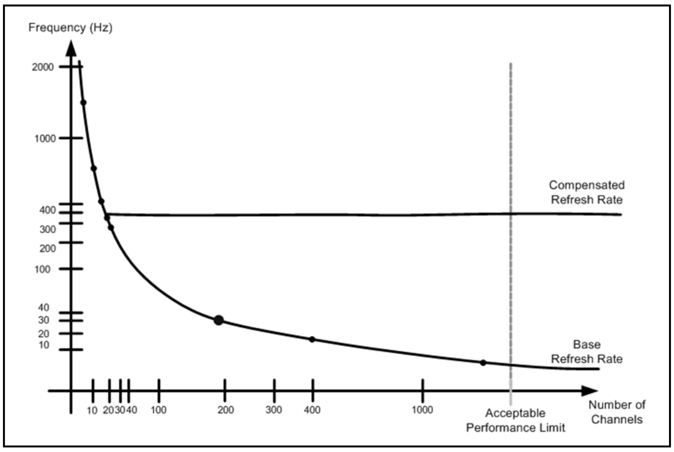


Fig. 17. Graphic of Channels-Frequency Trade-off

According to Fig. 16, the design experiment on A2F200 device to produce 48 channels PWM utilize 7.73% of FPGA resource. When this design is mapped to the smaller device A2F060, the FPGA fabric utilization for PWM generator is approximately 25.77%. In other word, 100% FPGA resource is theoretically capable of producing about 186 channels of PWM. In spite of 31.79 Hz base refresh rate of initial design, the use of accumulator shifts the refresh rate up to 380 Hz.

Adding more channel of PWMs reduces PWM base refresh rate gradually, still it can be compensated back to the specified minimum frequency (i.e. 300 Hz). In fact, the compensated frequency does not come to the real value when the counter value is below the multiplier frequency f^d. Hence, the consequence of low refresh rate still emerges in lower duty cycle setting. This side effect is negligible in case lower duty cycle might not noticeable. However, the more PWM channel are desired, the more multiplier frequency is needed. As a result, the more lower duty cycle value exist at lower frequency before the actual refresh is achieved. This phenomenon may cause side effect of lower frequency notable at lower duty cycle. The acceptable performance reaches its limit when the multiplier frequency is no longer capable of compensating the degraded PWM frequency. At this point, adding more channel further results in unsatisfying performance.

V. CONCLUSION

This paper has presented an architecture of multichannel digital PWM optimized for devices with embedded RAM blocks. By employing this method, the required logic element for the design is moved to the RAM. As a result, many more PWM channels are obtainable while maintaining output performance. The design is capable for parallelizing in case more RAM blocks are available. The design is available at hardware description language, thus portable to any digital semiconductor technology. The design was tested in Microsemi SmartFusion A2F200 Customizable System on Chip, giving the result of 7.73% FPGA usage to produce 48 channels of 380Hz 16 bits PWM.

REFERENCES

[1] M. Barr, "Introduction to Pulse Width Modulation," *Embedded Systems Programming,* p. 103–104, September 2001.

[2] T. Kugelstadt, "Design a Low-Cost PWM Circuit for Single IGBT-Drive Applications," Planet Analog, 2013.

[3] N. S. Corporation, *LM555 Timer Datasheet,* 2006.

[4] H. Meuth, I. Janiszewski and K. Schade, "Signalgenerator für pulsweitenmodulierte Signale auf rein digitaler Basis". Germany/München Patent DE 10 2005 032 672 A1, 9 February 2006.

[5] H. Meuth, I. Janiszewski and K. Schade, "Phase-Accumulator based Multi-Channel High-Precision Digital PWM Architecture," in *2005 IEEE International Frequency Control Symposium and Exposition*, Vancouver, 2005.

[6] Nexperia, *74LV595 Datasheet,* 2016.

[7] Microsemi, "SmartFusion Customizable System-on-Chip (cSoC)," Aliso Viejo, USA, 2012.

[8] "SmartFusion FPGA Fabric," Aliso Viejo, USA, 2011.