An Optimised Processor for FMCW Radar

Tim Styles
AptCore Limited
Bristol, UK

Leon Wildman
AptCore Limited
Bristol, UK

***Abstract*—This paper presents a digital signal processor architecture optimized for FMCW radar systems, as used in automotive, security and surveillance applications. The novel architecture is described, along with the size, power consumption and performance for key radar processing operations. Architecture features include a flexible compute unit optimized for FFT operations and a two-dimensional register file. An FPGA implementation of the processor is used to demonstrate range-Doppler processing in real-time.**

Keywords—Digital signal processors; Radar signal processing; Application specific integrated circuits

I. INTRODUCTION

Frequency Modulated Continuous Wave (FMCW) radar systems are becoming prevalent in automotive, security and surveillance markets, where they are used to measure position and velocity, and to classify detected objects. The low power nature of the technology enables compact and cost effective solutions for monitoring and detection in an increasing number of applications, from measuring traffic flow [1] to automatic emergency braking [2]. The all-weather, light level independent nature of millimetre wave systems makes them particularly suitable for applications where reliability is important, especially when visibility is impaired.

FMCW radar systems will be fitted as standard to an increasing range of vehicles, putting downward pressure on the cost of the radar system. Advances have been made in reducing the cost of the radar front end, including CMOS implementations [3] that enable tighter system integration. With the radar front end, modulation circuit, analog to digital converters and signal processor combined on a single chip, the cost of the system becomes highly dependent on the silicon area of the chip. An optimized signal processor reduces the size of the chip, resulting in a cost reduction for the overall system.

In high performance systems an optimized signal processor enables higher resolution and wider unambiguous range for distance and speed. The performance of the processor can be scaled to process data from multiple antennas, as used in MIMO radar systems, and the compact size of the processor allows multiple instances to be added to the system for redundancy and compliance with ISO 26262.

II. RADAR OPERATION

In a typical stepped FMCW radar system, the carrier frequency is modulated with a digitally generated linear ramp. At each step in the ramp, the received signal is demodulated and the phase is sampled. For a given step size, the change in phase of the received signal depends on the distance between the radar and the detected object. With equally spaced steps, the rate of change in phase, or frequency, is a function of the distance. A set of samples from a single ramp in the time domain can be transformed to the frequency domain to show the signal returned from each distance, known as the range profile [4].

Modulating the carrier frequency with a linear ramp is preferred, since it minimizes the bandwidth of the modulation signal to the voltage controlled oscillator in the radar front end, and it enables the use of a fast Fourier transform (FFT) in converting the measurements to the frequency domain.

If the distance to the object changes between samples, the frequency of the demodulated signal becomes a function of the distance and the rate of change in distance, or speed. This ambiguity can be resolved using multiple ramps, and several schemes have been described [5]. Sawtooth modulation, as shown in Fig. 1, consists of a set of identical ramps, where the samples for each ramp are converted to the frequency domain using a first FFT. The change in phase between each ramp is a function of the carrier frequency and speed of the object. With equally spaced ramps, this rate of change in phase, or Doppler frequency, can be determined using a second FFT.

Time domain samples from measurements using sawtooth modulation, can be converted to the frequency domain using a two-dimensional FFT. This technique is known as range-Doppler processing and each value in the two-dimensional result provides the signal amplitude and phase for each speed and distance.

In a system where multiple receivers are coherently sampled, the phase of the results can be used to determine the direction of arrival (DOA). A number of techniques exist for resolving multiple objects with the same range and speed [6].

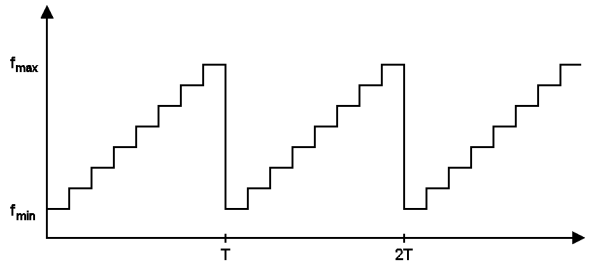

Fig. 1. Sawtooth modulation with 3 sweeps of period T, each with 8 steps from from f_{min} to f_{max}

III. PROCESSING

The first stage in processing the samples is to remove any brief periods of interference, typically from continuous wave transmitters. These occur when the carrier frequency matches the frequency of the interferer, resulting in a spike in the time domain of the received signal. A popular method of spike suppression detects saturation by comparing samples with upper and lower thresholds. If a threshold is exceeded, the sample is replaced with an estimated value based on neighbouring samples.

Fig. 2 shows the demodulated signal from a measurement using sawtooth modulation with 128 sweeps, each with 512 samples. The signal is dominated by low frequency components, but the signal of interest can be extracted providing the system has sufficient dynamic range.

A two-dimensional range-Doppler image is generated using a 2D FFT. The first FFT dimension generates a range profile from each of the ramps. Fig. 3 shows the stacked range profiles. A detection can clearly be seen at a range of 45 metres. The result is symmetrical since the processed samples are real.

The second FFT dimension generates a Doppler profile at each range using the range profiles from all of the ramps. Fig. 4 shows the resulting range-Doppler image. The detection at 45 metres has been resolved to a single object with a speed of 6 metres per second.

Before each FFT, the values are multiplied by a windowing function, which tapers the start and end of the sequence to zero, removing the discontinuities that would result in sidelobes around the peaks in the frequency domain. The data is transposed between the two FFT operations so that the storage can be accessed using the same optimized pattern.

Objects are detected in the range-Doppler image using a constant false alarm rate (CFAR) algorithm, where the complex magnitude of every value is compared to the local average, mitigating against changes in noise and wide band interference. This requires the magnitude to be calculated for every point in the range-Doppler image. Each local peak magnitude is then compared with the local average magnitude.

The operations for the described range-Doppler processing can be summarized as follows:

- Time domain spike suppression
- 128 x 512 point vector multiplication (windowing)
- 128 x 512 point fast Fourier transform
- 128 x 512 point transpose
- 512 x 128 point vector multiplication (windowing)
- 512 x 128 point fast Fourier transform
- 512 x 128 point complex magnitude
- Peak detection
- Calculation of local average
- Comparison of peak and local average (CFAR)

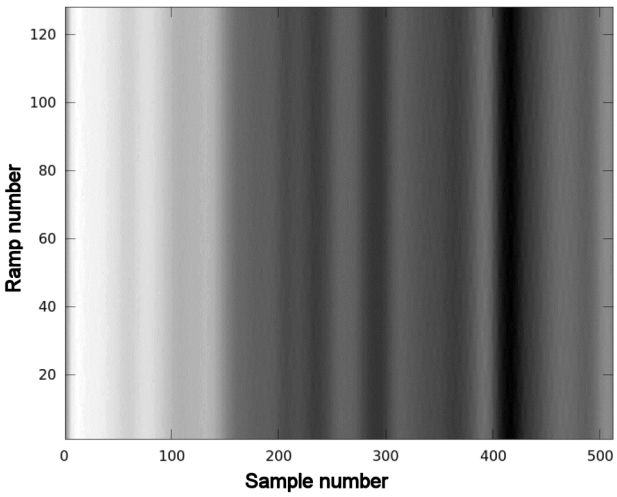

Fig. 2. ADC samples for 128 ramps each with 512 samples.

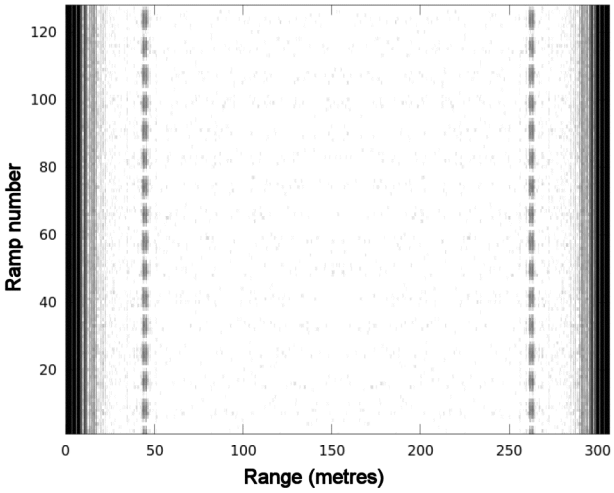

Fig. 3. Range profiles for 128 ramps with 0.6 metre range resolution. An object can be seen at 45 metres, with a time varying signal due to the speed of the object towards or away from the radar.

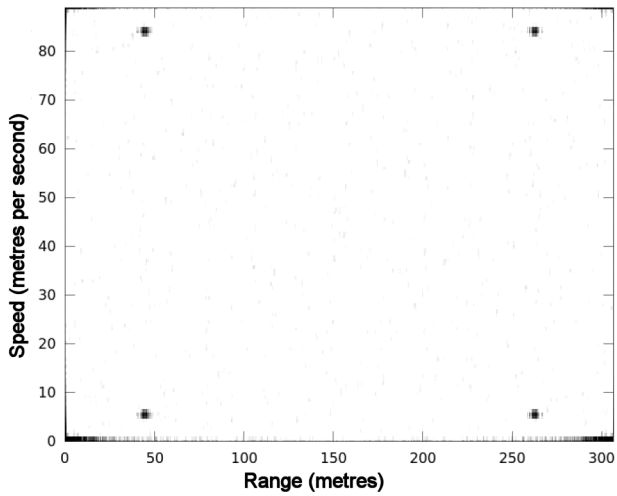

Fig. 4. Range-Doppler image showing a single object at a range of 45 metres with a speed of 6 metres per second.

IV. Processor Architecture

The AptCore ACR1 architecture, shown in Fig. 5, is centred on a flexible compute unit, shown in Fig. 6, which is optimised for FFT operations. In addition to this, it is flexible enough to perform a large number of other operations on parallel data [7]. The same unit is able to perform:

- Radix-2 butterfly
- Real or complex multiply
- Real or complex multiply-accumulate (MAC)
- Magnitude of complex number
- Maximum and Sum (used for CFAR)

The data for the compute unit is taken out of the register file, processed, and then returned to the register file. The register file is a dual bank two-dimensional array, similar to [8]. It enables efficient corner turning for matrix transpose, and address bit reversal for FFT data re-ordering. The two banks allow simultaneous transfer between the data memory and the compute unit.

A four register version of the register file is shown in Fig. 7. An index signal selects a row and column, and a second signal selects the indexed row or column. The AptCore ACR101 combines a four register version with one compute unit, while the ACR102 combines a sixteen register version with two compute units.

The constant memory contains the windowing function coefficients and the FFT twiddle factors. Any number of windowing functions can be stored subject to overall memory size. Since the addressing of the constant memory is programmable, it is possible to use the same twiddle factors for any sized FFT, up to the size of the constant memory.

Data is transferred between the bus interface and the data memory. As data is read in, it can be converted from unsigned to the internal 2's compliment signed format, and its size adjusted. In addition, the time domain spike suppression is performed using programmable thresholds and a programmable window around the spike.

As data is moved between the data memory and the register file, it can be block scaled to avoid overflow whilst retaining the largest dynamic range. The architecture of the register file enables multiple operations to be performed on the data, reducing the number of transfers to and from memory.

Cycle by cycle control of the data processing and movement is provided by a microcode programmable controller. Instructions come from either the host processor or a programmable set of macro functions. The macro functions are triggered by instructions from the host. In this way the host can provide fine level instructions for proprietary algorithms, while leaving common functions (FFT radix operations, for example) to the macro functions, reducing the instruction bandwidth.

The constant memory and microcode memory can be accessed from the bus interface. This needs to be done only once, although it could be repeated between different algorithms to reduce memory requirement.

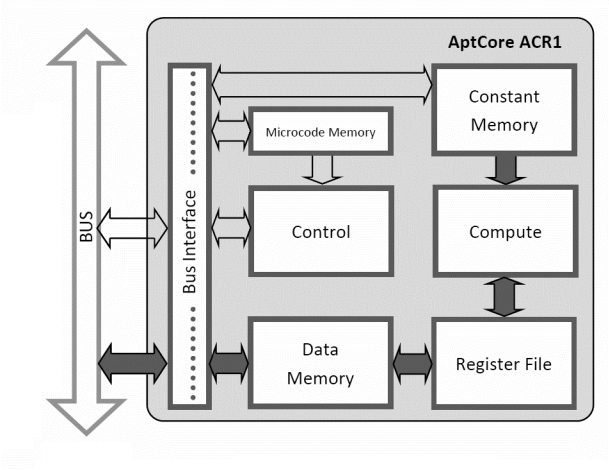

Fig. 5. ACR1 System diagram, showing the connections between the compute unit, the register file, and the constant and data memories. The microcode programmable controller is also shown.

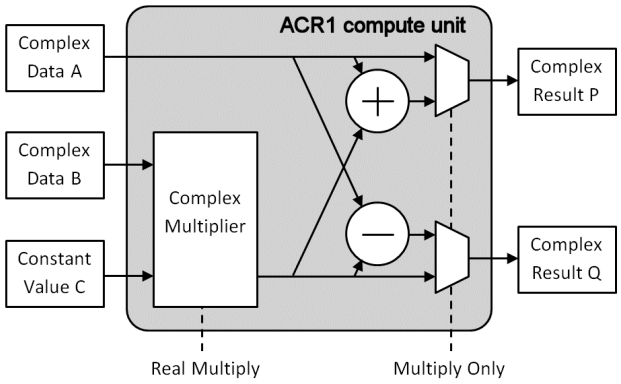

Fig. 6. ACR1 Compute unit, showing the two complex data inputs, the two complex data outputs, the complex constant input and the internal arithmetic units. The control signals that change the function of the units are also shown.

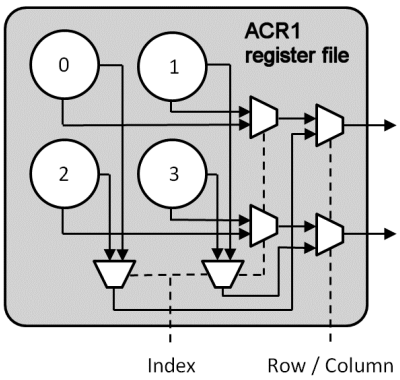

Fig. 7. A four register version of the ACR1 Register file, with two complex data outputs and the register selection logic. The complex data inputs and their associated logic are not shown.

V. PERFORMANCE

The ACR102 can perform four radix-4 FFTs in 8 cycles internally. Other operations can be performed at the rate of two complex values per cycle. TABLE I. gives the cycle count for windowing and FFT operations on data in the memory.

TABLE I. ACR102 PERFORMANCE

Data points	Cycle count		Total Time @120 MHz
	Windowing	*FFT*	
64	40	112	1.27 µs
128	72	240	2.60 µs
512	264	1176	12.0 µs
1024	520	2592	25.9 µs
4096	2056	12328	120 µs
8192	4104	26664	256 µs

The ACR102 has been synthesized for TSMC 65 nm GP using a regular VT library, excluding RAMs. At 120 MHz the expected area is 60k NAND equivalent gates, and the dynamic and leakage power consumption figures are 0.8 mW and 0.4 mW respectively.

VI. FPGA IMPLEMENTATION

The ACR102 has been implemented in the field programmable gate array (FPGA) logic of a Xilinx Zynq device. The device includes a dual core ARM processor, dual ADC inputs, and various interfaces including CAN bus and Ethernet.

A number of radar front end (RFE) units were tested with the system, including a 24 GHz module from InnoSent. Fig. 8 shows the result of range-Doppler processing the measurement of a pedestrian and a car. The car is at a range of 45 metres and the pedestrian is at a range of 10 metres. The pedestrian is jogging at about 2 m/s and the car is moving at 2.5 m/s away from the radar. The car appears as a single return, while the pedestrian return contains a central velocity component and components either side corresponding to body parts moving slower than the body and faster than the body.

The radar was configured to complete a set of sweeps every 40 milliseconds. The 2D FFT took 10 milliseconds with the FPGA logic clocked at 35 MHz. The ARM core was used to transfer the magnitude values to a PC for display. The ARM core could also perform CFAR detection by comparing the detected peaks with their local average.

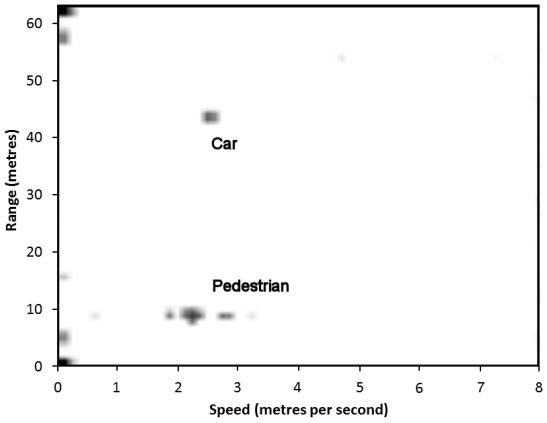

Fig. 8. Range-Doppler measurement of a pedestrian and car. The car can be seen at a range of 45 metres and speed of 2.5 metres per second. The pedestrian can be seen at a range of 10 metres and a similar speed. Some micro-Doppler signals from the pedestrian can also be seen.

VII. CONCLUSION

The AptCore ACR1 DSP architecture is optimised for radar applications. The novel compute unit and register file, together with wide internal data-paths and low data movement, provide an inherently low power and small architecture. This combined with the low clock speed requirement results in an extremely area and power efficient design, while still providing the flexibility of a fully programmable solution. The architecture is ideally suited to safety critical, low power radar and related signal processing.

The ACR102 has been implemented in a Xilinx Zynq device and demonstrated with various system configurations, including range-Doppler processing. The system was tested with various radar front end units, and measurements were taken of vehicles and pedestrians.

REFERENCES

[1] Hao Zhang, Yu Wen, and Sun Xiaowei, "Adaptive traffic lane detection based on normalized power accumulation." Intelligent Transportation Systems, 2008, 11th International IEEE Conference on. IEEE, 2008.

[2] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking." Intelligent Transportation Systems, 2010, 13th International IEEE Conference on. IEEE, 2010.

[3] T. Mitomo, N. Ono, Y. Yoshiara, O. Watanabe, and I. Seto, "A 77 GHz 90 nm CMOS transceiver for FMCW radar applications," IEEE J. Solid-State Circuits, vol. 45, no. 4, pp. 928–937, Apr. 2010.

[4] A. Stove, "Linear FMCW radar techniques," Radar and Signal Processing, IEE Proceedings F, vol. 139, no. 5, pp. 343 –350, oct 1992.

[5] V. Winker, "Range Doppler detection for automotive FMCW radars," European Radar Conference, pp. 166-169, Oct. 2007.

[6] R. H. Roy and T. Kailath, "ESPRIT—estimation of signal parameters via rotational invariance techniques," IEEE Trans. Acoust., Speech, Signal Process., vol. 37, June 1989.

[7] T. Styles, J. Hutchinson, "Signal processing apparatus," U.K. Patent GB2490914, Nov. 21, 2012.

[8] Yu-Wei Lin, Hsuan-Yu Liu, and Chen-Yi Lee, "A Dynamic Scaling FFT Processor for DVB-T Applications," IEEE Journal of solid-state circuits, vol. 39, no. 11, November 2004.