

Implementation of a High Speed Four Transmitter Space-Time Encoder using Field Programmable Gate Array and Parallel Digital Signal Processors

Peter J. Green and Desmond P. Taylor

Department of Electrical and Computer Engineering University of Canterbury

Christchurch

New Zealand

peter.green@canterbury.ac.nz

taylor@elec.canterbury.ac.nz

Abstract

This paper describes the concept, architecture, development and demonstration of a high performance, 4 transmitter, real-time space time encoder designed for research into transmitter diversity and multiple input and multiple output (MIMO) wireless systems. It is implemented on a Xilinx Virtex 2 Pro Field Programmable Gate Array (FPGA) and parallel processing on multiple Freescale DSP56321 digital signal processors (DSP). The system is software defined to allow for flexibility in the choice of transmit modulation formats, data rates and space-time coding schemes. Hardware, firmware and software aspects of the space time encoder system to meet design requirements are discussed. The testing and demonstration of the system running the Alamouti space time coding scheme is covered.

The current implementation is an enhancement to an existing Smart Antenna Software Radio Test System (SASRATS) platform [3, 4] designed to test and verify various space time architectures and algorithms. Of significant interest is the real-time testing of the space time (ST) coding schemes developed by Alamouti [1] and others mentioned in [2]. Space time coding schemes are necessary to support the high data rates of future *wireless* mobile and local area network standards. The primary objective is to increase system capacity and performance through the use of multiple antennas, spatial multiplexing and space time (ST) coding.

A requirement for real-time space-time coding experiments is that all transmitters must be synchronised. Each transmitter must output data from the space-time encoder algorithm at precisely the same time. Our original transmitters, developed in 2000 [3] were designed for beamforming experiments and not for synchronised space-time encoding operations. Our goal now is to achieve synchronised transmit symbol rates of greater than 1 Mbaud per transmitter

with pulse shaping from 4 transmitters. Another requirement is that the transmitter characteristics must be *software defined* to allow for flexibility in choice of modulation formats, data rates and space-time coding schemes.

To meet the desired TX data rates and programmability objectives; the Xilinx Virtex 2 Pro FPGA, Motorola DSP56321 DSPs' and Analog Devices quadrature digital upconverter AD9857 integrated circuit (IC) were selected. The AD9857 IC has an integrated direct digital synthesizer, 14-bit digital-to-analog converter and quadrature modulator. The AD9857 IC operates at a clock frequency of 200 MHz and is programmed to output a 70 MHz intermediate frequency (IF) signal. This is then upconverted to 915 MHz by a separate SASRATS analog radio frequency (RF) upconverter unit. The required symbol rate is programmed into the AD9857 and the device generates an output clock (PDCLK) signal at twice the symbol rate. Once enabled for quadrature modulation, the device will request 14 bit In phase (I) and Quadrature phase (Q) signals. The IQ signals must be presented sequentially and continuously to the AD9857 and clocked on the rising edge of the PDCLK. After an IQ pair is received, the 70 MHz modulated signal is produced.

The complete system implementation consists of a master Xilinx Virtex 2 Pro FPGA, 4 slave DSPs' and 4 sets of the AD9857 Upconverter boards as shown in Figure 1. The four DSPs' are used to handle the computation overheads of pulse shaping for each transmitter. The FPGA performs random number generation, mapping data into the desired digital modulation format and space time encoding. Data is transferred to the 4 slaves in parallel through four 16-bit ports configured on the Xilinx FPGA board. The data on the FPGA output ports are distributed to each Slave DSP through their respective Port A's triggered by interrupt driven Direct Memory Access (DMA) transfers. On each slave, the finite impulse response (FIR) filtering is carried

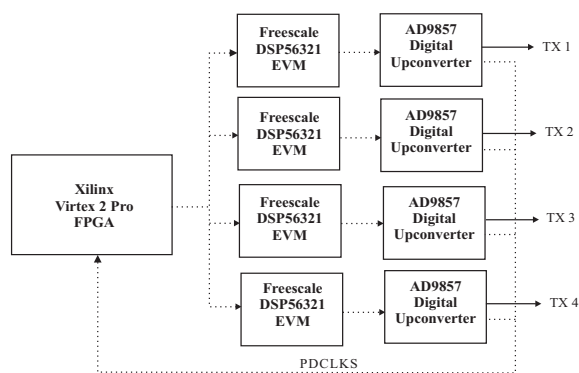


Figure 1. SASRAT 4 transmitter space-time encoder

out by the enhanced filter co-processor (EFCOP). The filtered data is then sent to Port B. Port B is a 16 bit port and has sufficient resolution to output the 14 bit data required by the AD9857 TX boards. Timing is controlled by the PDCLK signals from the AD9857 TX board to the master FPGA. The master then sends Interrupt Requests (IRQ) to the slave DSPs at the appropriate time instant for transfer and FIR processing of data.

1 Pulse shaping of transmitter symbols

One requirement of any digital transmitter is the need for pulse shaping filters. This is to shape the transmitted spectrum to meet out of band emission requirements and ensure that at the receiver, the received signal is sampled at an optimal point in the pulse interval to maximize the probability of an accurate decision. The symbol pulses must not interfere with one another at the optimal sampling point. A rectangular pulse can be used but is not ideal as it takes infinite bandwidth. Raised cosine pulse shaping is normally used between transmitter and receiver to conserve bandwidth and to ensure no intersymbol interference at the sampling points. The filters are implemented using finite impulse response (FIR) filters on the DSP. However, to ensure that the raised cosine frequency characteristics are met, the filter must oversample the data by at least a factor of 2 (samples per symbol).

The sequential data input format to the AD9857 and minimum pulse shaping (2 X oversampling) requirements imply that for a transmit symbol rate of 1 Mbauds, new data must be calculated and presented to the AD9857 at 4 MSPS per transmitter and this requires a fast digital processing platform. To provide a more accurate spectral shape, it is also desirable to oversample by a factor greater than 2. This requirement will increase the FIR filter length significantly.

The design allows for filter lengths of up to 512 taps.

The high speed pulse shaping function can be directly implemented on the FPGA platform but is limited by the number of 18×18 bit multipliers available in the various versions of the Virtex 2 Pro. At the time of writing, the best version (2VP100) of the Virtex 2 Pro has 444 18×18 bit multipliers. The number of cells in our Virtex 2 Pro (2VP30) is limited to 136 multipliers. Using the 2VP30 with 8 parallel (IQ) processing paths for 4 transmitters leaves only 17 multipliers. This gives a 16 tap FIR filter per path. Another option is to have the I and Q data time-multiplexed on 4 processing paths to give a 33 tap FIR filter/per path. Adding clever reuse of multipliers and coefficients in a symmetrical coefficient FIR filter design can have the equivalence of a 66 tap FIR filter. This is still below the goal of a 512 tap FIR filter. For this reason, we implement the pulse shaping function on the four DSP56321's.

The DSP56321 has an enhanced filter co-processor (EFCOP) which can be configured for FIR filtering. The EFCOP has 12K-word data and 12K-word coefficient memory banks and can easily implement a 512 tap FIR filter at the required rate. The DSP56321 is configured to perform DMA transfer from the FPGA to the EFCOP on the negative edge of the IRQ signal through Port A and outputs the filtered sample from the EFCOP to Port B. The data on Port B is read by the AD9857 TX boards on the rising edge of the IRQ.

However, there is a disadvantage to this approach !!. A factor that limits the data rate when using the DSP56321 is the processing speed of the interrupt service routine. The DSP does not respond instantly. There is a significant time delay of 50 nS between the detection of the negative edge of the IRQ signal to the first execution of required instructions as the DSP takes clock cycles to set up the stack and other registers to respond to an interrupt service routine. It takes a further 60nS to process and output data onto Port B. Another 10nS guard time is added to ensure that data is stable on the rising edge of PDCLK to bring the total time to 120nS. Thus a period of 480 nS (2.082 MHz) is needed to send an IQ sequential pair to the transmitter. If the DSP outputs 2 samples per symbol, then the achievable output symbol rate from the transmitter is just above 1 Mbauds.

If pulse shaping were done by the FPGA, a much higher output symbol rate could be achieved as the FPGA approach does not incur any IRQ and DMA interrupt overheads etc. All processing is done on dedicated multipliers. The limitation of the FPGA is the small number of filter taps. Thus for applications requiring very high speed but short filter lengths, the FPGA approach is recommended. For these applications, the hardware design allows the FPGA to bypass the DSPs and connect directly to the AD9857 upconverters.

Table 1. Example of the 2 transmitter Alamouti scheme over 6 symbols

	Time Instants					
	t	$t + T$	$t + 2T$	$t + 3T$	$t + 4T$	$t + 5T$
TX0	s_0	$-s_1^*$	s_2	$-s_3^*$	s_4	$-s_5^*$
TX1	s_1	s_0^*	s_3	s_2^*	s_5	s_4^*

2 Alamouti encoding and its implementation on FPGA

In a classical one-transmitter system, symbols s_0, s_1, s_2, \dots are transmitted at times $t, t + T, t + 2T, \dots$ respectively. In a two transmitter Alamouti encoder scheme however, the symbols s_0 and s_1 are transmitted simultaneously from transmitters TX0 and TX1 at time instant t . At time instant $t + T$, the symbols $-s_1^*$ and s_0^* are transmitted simultaneously out of the transmitters where * represents the complex conjugate. Table 1 shows the transmitted symbols for a 6 symbol 2 transmit Alamouti scheme.

The complete design is implemented using schematic entry on the *Xilinx Integrated System Environment (ISE) Foundation* design tool. ISE has a large library of functional blocks such as adders, multipliers, registers, memory and logic for schematic entry. VHDL code can also be integrated as a block with other schematic components if desired. This approach allows hardware designers to quickly use FPGA technology to implement hardware designs without mastering VHDL. The ISE tool then translates the design into firmware that is needed to program the Virtex 2 Pro. The ISE tool also incorporates the *Xilinx Core Generator* intellectual property(IP) modules with functions such as FIR filters which can be embedded into a schematic design to shorten design cycle time.

The system architecture to implement a real time, continuously operating 2 transmit Alamouti scheme is shown in Figure 2. It consists of the quadrature phase shift keying (QPSK) random generator block, the clock generators (CR8CE), the register latch (FD4CE), the Look Up Table (Alamouti_LUT4.V3) block with tri-state output buffers (OBUFT) and the transmit enable controller (ILD).

In the random generator implementation, the symbols are QPSK symbols where each symbol represents 2 bits of data. First, a pseudo random sequence generator is designed to generate the random bits. Figure 3 shows the implementation of a 24 bit maximal shift random generator which consists of a concatenation of an 8 bit(SR8RLE) and 16 bit (SR16RLE) programmable shift registers taken from the Xilinx library. A feedback signal is derived from specific tap points in the shift registers via a three input EXOR gate. The registers must initially be reset and loaded with a preset 24-bit data which acts as a seed in the random number generator. Once loaded and enabled, the generator will output

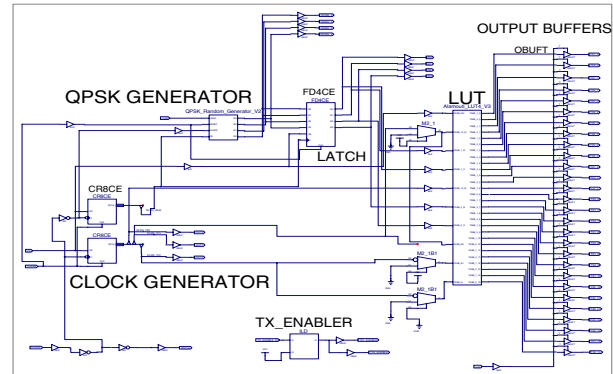


Figure 2. Schematic of a 2 transmitter space-time encoder implemented on Xilinx ISE

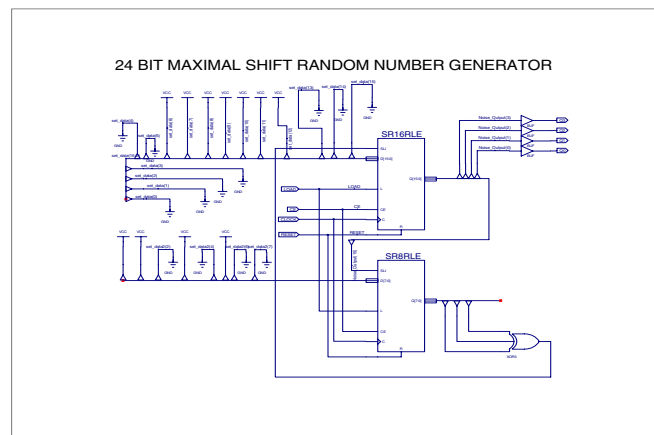


Figure 3. QPSK Random Number Generator

a random bit at the rising edge of each clock cycle. To generate 2 parallel paths for a 2 transmit Alamouti scheme, 4 bits of data (Q0, Q1, Q2, Q3) are tapped out of the random generator shift registers to generate 2 QPSK symbols. The 4 bits are stored in a register (FD4CE) in Figure 2 for further processing. A fresh set of 4 bits is latched after every 4 clock cycles.

The clock generator circuitry ensures that all signals are clocked and latched at the correct instant in time. The clock input is derived from the PDCLK signal from the AD9857 transmitter board. Note that the I and Q signals must be presented in sequence at the rising edge of PDCLK and repeated indefinitely until the TX_ENABLE is disabled. The PDCLK is first inverted to drive the system and then inverted again to output as an IRQ signal to the DSP's. Two binary ripple counters (CR8CE), one positive and the other negative edge triggered, are used to generate the various clock signals. The various clock signals generated by the clock circuitry are shown in Figure 4. CLK3 enables the

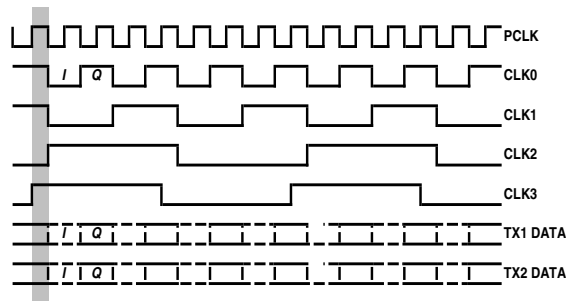


Figure 4. Clock signals

the Random Generator over 4 PDCLK cycles to allow a fresh set of 4 bits to be generated. CLK0, CLK1, CLK2 and latched values of Q0, Q1, Q2 and Q3 are address lines to the Look Up Table (LUT) circuit block. CLK0 when '0' indicates an I data output, Q data output when '1'. CLK1 remains '0' or '1' for the duration of one IQ pair (i.e. one time slot). CLK2 remains '1' or '0' for the duration of 2 IQ pairs (2 time slots).

Q0 and Q1 determine the QPSK symbol to be output to transmitter TX1 and Q2 and Q3 set the symbol data to TX2. Q0, Q1, Q2 and Q3 remain unchanged over 4 timeslots where each timeslot consist of an I and Q data sequential pair (AD9857 requirement). Although only 2 timeslots are required for Alamouti, the extra two are added for requirements of the pulse shaping filters (2X oversampling) in the DSP56321 Coprocessor. Thus the original Alamouti symbol sequence from TX1 is changed from $s_0, -s_1^*$ to $s_0, 0, -s_1^*, 0$. Similarly, the symbol sequence from TX2 is changed from s_1, s_0^* to $s_1, 0, s_0^*, 0$. Therefore to maintain the original symbol rate, PDCLK must be increased by a factor of 2.

The Alamouti_LUT4 block consist of 2 LUT banks (BANK 1 and BANK 2), each of size 2 X 16 X 16 as shown in Figure 5. There are 2 sets of 16-bit words per bank (for TX1 and TX2) and there are 16 words in each bank. Each word has a unique address. The outputs of the two banks are multiplexed using 32, 2-input multiplexers (M2_1E) under the control of CLK2. When CLK2 is '0', the outputs to TX1 and TX2 comes from BANK 1 in the first and second timeslots. During the third and fourth timeslots when CLK2 is '1', the outputs to TX1 and TX2 come from BANK 2.

The circuitry of each LUT bank is shown in Figure 6. Each LUT bank is made up of 16 cells. Each cell is made up of two 4-bit LUTs' (LUT4) and two multiplexers (M4_1E). Each cell has two outputs, each representing one bit of a 16-bit word for TX1 and TX2. The outputs of each LUT4 in a cell is sent to two multiplexers (M4_1E) which select

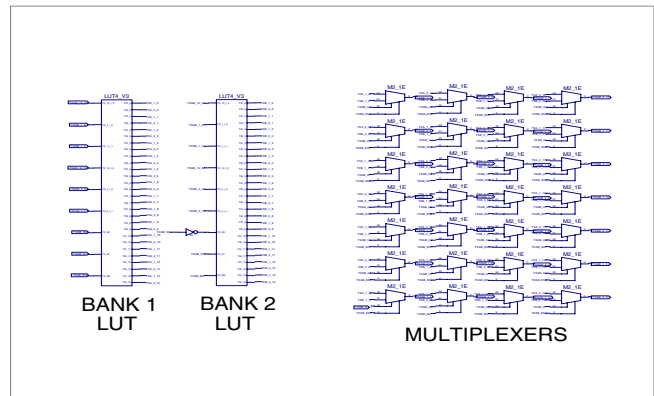


Figure 5. Structure of 2 Banks of Look Up Tables with multiplexed outputs

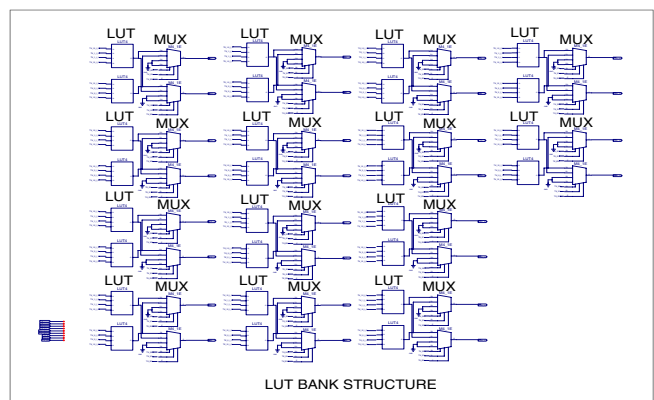


Figure 6. Internal structure of one LUT bank

the correct data to send to the transmitters TX1 and TX2. The multiplexer is needed because in the Alamouti 2 TX scheme, the second symbol at the first transmitter is the negative conjugate of the first symbol of the second transmitter. Similarly the second symbol at the second transmitter is the conjugate of the first symbol of the first transmitter. The switching of the multiplexer outputs is controlled by CLK1.

The values prestored in the LUTs' for the Alamouti 2 transmit QPSK scheme with 2 X pulse shaping filter oversampling requirement are shown in the tables of Figure 7. Other data formatting options and space-time codes can be configured by programming the LUTS with the appropriate data and the properly setting the LUT control lines. The overall design is then verified on a simulator platform from Mentor Graphics called ModelSim XE-III. The simulator enables the verification of the HDL source code and the functional and timing models generated by the ISE Foundation software.

CLK(2)	Q(0)	Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	Q(6)	Q(7)	Q(8)	Q(9)	Q(10)	Q(11)	Q(12)	Q(13)	Q(14)	Q(15)
TX1 BANK 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TX2 BANK 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TX1 SLOT 3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TX2 SLOT 4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7. LUT data for 2 transmitter QPSK modulated Alamouti scheme

2.1 Software algorithm for DSP slaves

When data is valid at the outputs of TX0 and TX1, the FPGA sends an IRQ signal to the DSPs'. The DSPs' respond on the falling edge of the IRQ signal and perform DMA transfers from the TX0 and TX1 outputs to the enhanced filter coprocessor (EFCOP) on the respective DSPs' to perform pulse shaping of the symbol as shown in the flowchart of Figure 8. On completion of the FIR computation, the filtered I sample is sent to Port B. This filtered sample is loaded into the AD9857 upconverter on the rising edge of the PDCLK signal. On the next falling edge of the IRQ signal, the Q sample is processed in a similar manner. When both I and Q samples are loaded into the upconverter, the IQ data modulates the 70 MHz intermediate frequency. This process continues indefinitely until the TX.Enable control line is disabled. Note that there is a period of latency from the moment the TX.Enable control line is enabled to the first valid data from the transmitter. This latency period is dependent on the length of the FIR filter programmed into the EFCOP. A Data_Valid bit on the EFCOP register is monitored at start up.

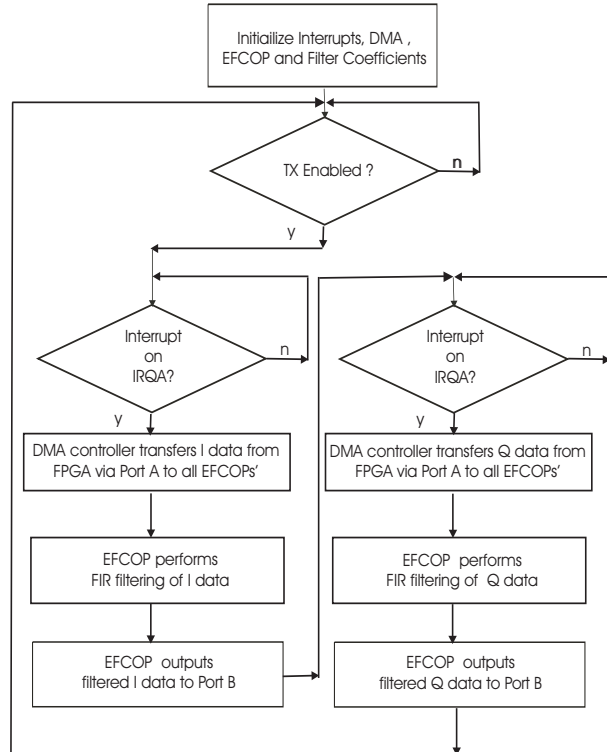


Figure 8. Flowchart of the DSP56321 slave software

2.2 Testing and system performance

On testing the system, it was found that the PDCLKs of the four AD9857 upconverters synchronize at different phases of the PDCLK waveform. In the initial design, the four upconverters ran from a common 10 MHz clock. Each AD9857 has an internal digital phase lock loop (DPLL) circuit that synthesizes a 200 MHz internal clock from the 10 MHz reference source. However, it is found that the each AD9857 locks to 200 MHz at slightly different times and thus it is impossible to get the PDCLK signals from all 4 units to align precisely. To resolve this problem and maintain perfect phase alignment among four AD9857 ICs in this specific application, we choose to bypass the internal DPLL and run a common 200 MHz reference source to all units. This is achieved in the final design by using a CDC111 IC from Texas Instruments which can produce up to 9 synchronized 200 MHz differential outputs from a common 200 MHz clock source. Four differential outputs are used to drive the four AD9857 upconverters.

The space time encoder is fully commissioned and the set-up is shown in Figures 9 and 10. The encoder is housed in a separate chassis to minimize interference between the

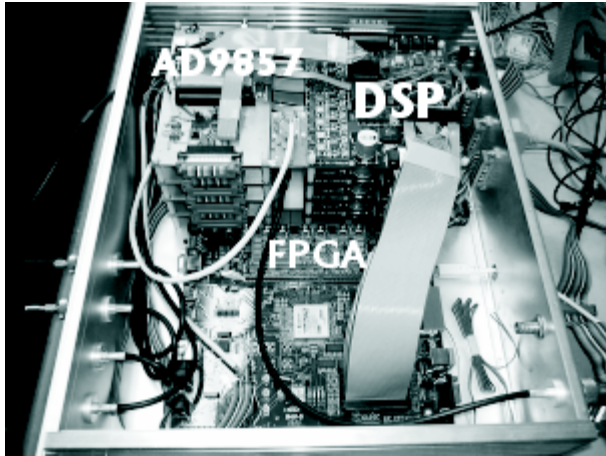


Figure 9. Space time encoder hardware for 4 antenna system

digital encoder circuitry and the analog RF upconverters. The four AD9857 modulator boards and the four DSP slaves are stacked one above the other primarily to minimize interconnect lengths for high speed data transmission among boards and also to conserve space in the chassis. The 4 transmit system has been tested using BPSK and QPSK modulation and optimized to 1.5 MBauds symbol rate per transmitter using 2X oversampling and FIR filtering on the DSP. A 2 transmitter QPSK Alamouti encoder scheme has been fully tested and a 4 transmitter orthogonal space time code will be implemented in the near future. The system is limited by the interrupt and program processing speed of the DSP slaves. However the system has been tested to operate up to 5 MBauds symbol rate using direct connection between the FPGA and AD9857 upconverters but with limited pulse shaping. Symbol rates are limited not by FPGA speed but by the surface acoustic wave (SAW) filter bandwidth of 10 MHz used in the analog upconverters. The SAW filters are used in the analog upconverter circuitry to limit bandwidth and control unwanted spurious emissions of the radio spectrum at 915 MHz.

3 Conclusions

We have described the design, development and successful implementation of a 4 transmitter space time encoder based on a Xilinx Virtex 2 Pro FPGA board, Freescale DSP56321s' for pulse shaping and the Analog Devices AD9857 modulator boards are capable of carrying out space time coding algorithms of up to 4 transmitters. The encoder is fully operational and a 2 transmit Alamouti scheme has been implemented and tested. The system is fully soft-

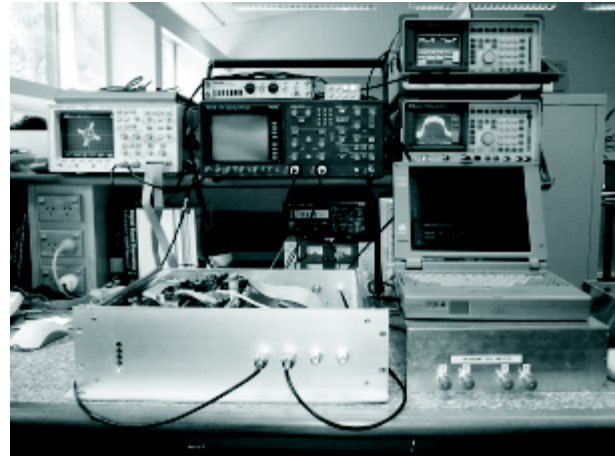


Figure 10. Complete SASRATS transmit platform with space time encoder

ware programmable and allows changes to be easily made by changing the firmware on the FPGA or software in the DSP slaves.

References

- [1] S. Alamouti. Space block coding: A simple transmitter diversity technique for wireless communications. *IEEE J. Select. Areas. Communication*, 16:1451–1458, Oct. 1998.
- [2] D. Gesbert et al. From theory to practice: An overview of mimo space-time coded wireless systems. *IEEE Journal on Selected Areas in Communications*, 21:281–302, Apr. 2003.
- [3] P. Green and D. Taylor. Smart antenna software radio test system. *Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications.*, 1:68–72, Jan. 2002.
- [4] P. Green and D. Taylor. Experimental verification of space-time algorithms using the smart antenna software radio test system (sasrats) platform. *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, 4:2539–2544, 2004.