

# Modelling and simulation of wireless sensor system for health monitoring using HDL and Simulink<sup>®</sup> mixed environment

K. Arshak, E. Jafer, D. McDonagh and C.S. Ibala

**Abstract:** The aim of this study is to model and design an efficient wireless system that is easy to integrate with other technologies or infrastructures at a low cost. The system would read analogue information recorded by a biomedical sensor in a transmitting unit attached to the patient. The recorded data are converted digitally using analogue-to-digital converter and sent to frequency-shift keying (FSK) transmitter through field programmable gate arrays (FPGAs). Verilog-HDL has been used to implement the required functions of the FPGA, such as bus interfacing, data buffering, compression and data framing. On the other hand, Simulink<sup>®</sup> software has been used to model and simulate FSK transmitter/receiver architecture suitable for short-range communications. Basically, a two-tone FSK signal is generated and passed through a noisy channel, which is then downconverted to baseband and passed to the frequency-modulating detector to restore the original transmitted bit stream. These illustrate how easily the mixed signal modelling can be well mapped into hardware description language (HDL) and mathematical programming techniques. The developed simulation models are used to explore the design change options. The behavioural HDL design has been interfaced to the Simulink model using system generator in a co-simulation environment, and the overall performance has been verified.

## 1 Introduction

Home healthcare costs have increased by 400% in the last five years. Although hospital or nursing home care can be delivered at the patient's home, professional services such as clinical and medication monitoring are still required. On the other hand, advancement in wireless and Internet technologies are developing rapidly and has opened new opportunities for the health services to reconsider the traditional model of patient care [1]. In the next 25 years, the global population over the age 65 would increase by 88% [2].

The challenge is to raise or at least maintain the present level of healthcare provision without ending up in an uncontrolled cost explosion. The increasing number of researchers and manufacturers who are working towards a new generation of wireless technology applications in the field of medicine has led to an improvement in the quality of patient care and also reduction in cost [3]. One of the areas in healthcare that best lend itself to wireless technology is patient monitoring, also known as wireless telemetry. By attaching a wireless monitoring device to the patient's bedside or directly to the patient, physicians can monitor vital signs from down the hall to across the town. The most critical features of wearable health monitoring

equipment are long battery life, lightweight and small dimensions [4]. Continuous monitoring during normal activity also requires that the device can be hidden to protect user's privacy. In order to make the monitoring devices easily wearable and suitable for the patient, they need to be designed ergonomically. The equipment must also produce high-quality signals, be easy to wear and remove, robust and suitable for different environments and users, both males and females [5].

The aim of this work is to present a useful model that combines both hardware and software environments and achieves the described demands. The key features of the introduced models are low complexity, low power consumption and efficient data transmission. The main motive behind using a filed programmable gate array (FPGA) device to implement the digital part of the model is to have a ready platform for ASIC, which is the next stage of this research work. In addition, FPGA design is characterised to be configurable and can be upgraded any time, which is an attractive feature that cannot be obtained with any other fixed hardware microprocessors or digital signal processing (DSP) devices.

## 2 Basics of mixed simulation

In the design process of heterogeneous systems, different languages can be used to study the different subsystems. Mixed simulation is based on a distributed simulation approach allowing to validate the whole set of specifications at different stages of the design process [6]. Fig. 1 shows the different stages of the software and hardware designs and shows how different languages can be used ('Matlab' and 'HDL' codes). There are in fact two types of mixed simulation: the first one allows validation of the specifications

© The Institution of Engineering and Technology 2007

doi:10.1049/iet-cdt:20050206

Paper first received 1st December 2005 and in revised form 27th March 2007

K. Arshak and E. Jafer are with the Electronic and Computer Engineering Department, University of Limerick, Plassey Technological Park, Limerick, Ireland

D. McDonagh is with Toumaz Technologies Limited, Abingdon Oxfordshire, OX143DB, UK

C.S. Ibala is with XILINX, CAE Logic Drive, Dublin, Ireland

E-mail: khalil.arshak@ul.ie

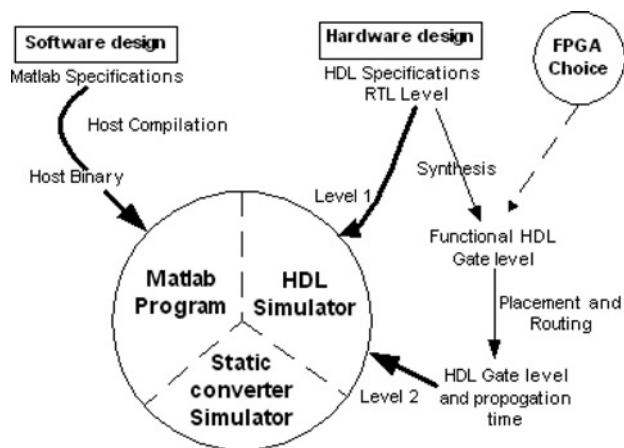


Fig. 1 Mixed simulation in the design cycle

without taking the temporal limitations of a given technology into account (level 1, Fig. 1). The second co-simulation level allows the simulation of the digital control including some properties of the technologies used. With this type of co-simulation, it is possible to check whether some of the constraints, like computation times, accuracy and propagation times (for FPGAs) are respected. The mixed simulation environment that is used involves the two domains that are discussed above. The two FPGA designs have interfaced efficiently with the RF Simulink models using system generator (Sysgen) tool that acts here as the converter simulator.

### 3 System generator (Sysgen) for co-simulation

Sysgen from Xilinx [7] has been developed to provide the capability of modelling and implementing high-performance DSP systems in FPGAs in a Simulink environment. It can automatically translate a DSP system developed using Matlab and Simulink into highly optimised hardware description models. In addition to the availability of the simulation libraries for high-level modelling and automatic code generation, Sysgen provides a high-speed hardware description language (HDL) co-simulation interfaced to the ModelSim simulator.

### 4 Modelling using Simulink® and Verilog-HDL

Earlier, Simulink has been used successfully to explore different system-level models for wireless transceivers. An application in utilising the communication toolbox is illustrated in [8]. The Simulink macro-models include the quadrature up/down converter with quadrature local oscillator [9–11] and double-quadrature up/down converter with passive quadrature generators [12, 13]. Other models have been developed to study some problems associated with these architectures, such as mismatch and synchronisation [14].

Verilog-HDL has been employed to develop both behavioural and hardware synthesisable models. The FPGAs, that contain programmable logic components and programmable interconnects devices are good tools to test and verify the functionality of the developed models [15]. FPGAs are based on flexible regular structures, which upon configuration can emulate both sequential and combinational logic circuits. These flexible structures contain four major types of blocks: the logic block, the interconnect structure, the connection block and the input/output (I/O) block. Recently, a wide scale of applications has been presented based on Verilog models, such as data acquisition systems

for health monitoring [16], custom design processors and wireless systems [17, 18].

Little work has been done so far to investigate the benefits of having mixed environment simulation models. In [19], a new tool for rapid prototyping FPGAs in hybrid systems was developed using Simulink as a high-level description language in a flexible simulation environment. Another approach based on top-down mixed signal modelling was presented in [20]. In this approach, Simulink was used first to build the behavioural model. Sysgen addresses the demand for such co-simulation with highly efficient simulator interfaces and the environment ready for plugging-in third party tools. The work in this paper uses the Simulink interface feature delivered with Sysgen, which allows to link with any HDL model.

## 5 System specifications

The main blocks of the system are sketched in Fig. 2. First, an eight-bit ADC handles the analogue signal recorded by the sensor readout circuit. The FPGA then makes different processing on the digital data like buffering, compression and framing, and then send the data to the FSK transmitter model. Behavioural models for both transmitter/receiver have been implemented using Simulink based on the direct conversion architecture [21].

A novel logic detector has been modelled at the receiver side to recover the transmitted symbols. A second FPGA is interfaced to the output of the receiver and is responsible for processing the data inversely like de-framing and de-compressing. A detailed description of the HDL model is presented in Section 6. Sysgen interface has been used to invoke the HDL modules in the Matlab and Simulink environment.

## 6 FPGA description

The main blocks of the transmitter side FPGA are shown in Fig. 3. The different units of the system were coded using Verilog-HDL simulated with ModelSim SE V6.0a and implemented using ISE7.1. The final implementation was done using Spartan-3 device, as it provides the various features that solve designer's challenge throughout the entire system. From Fig. 3, the FPGA consists of mainly a serial peripheral interface (SPI), run length encoding (RLE) compressor and high data link control (HDLC)

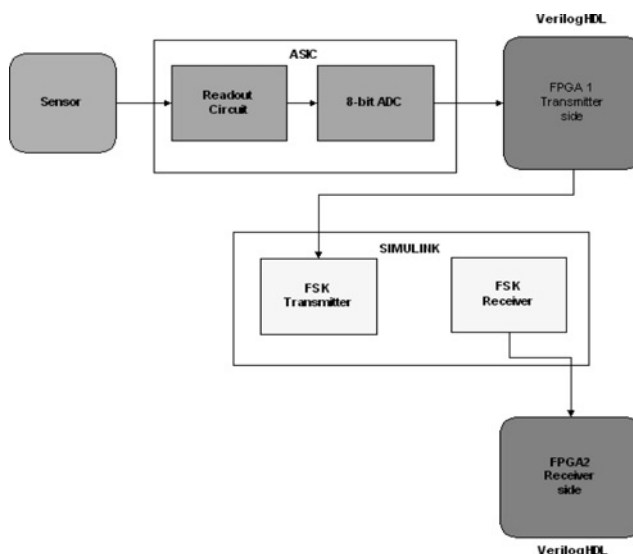
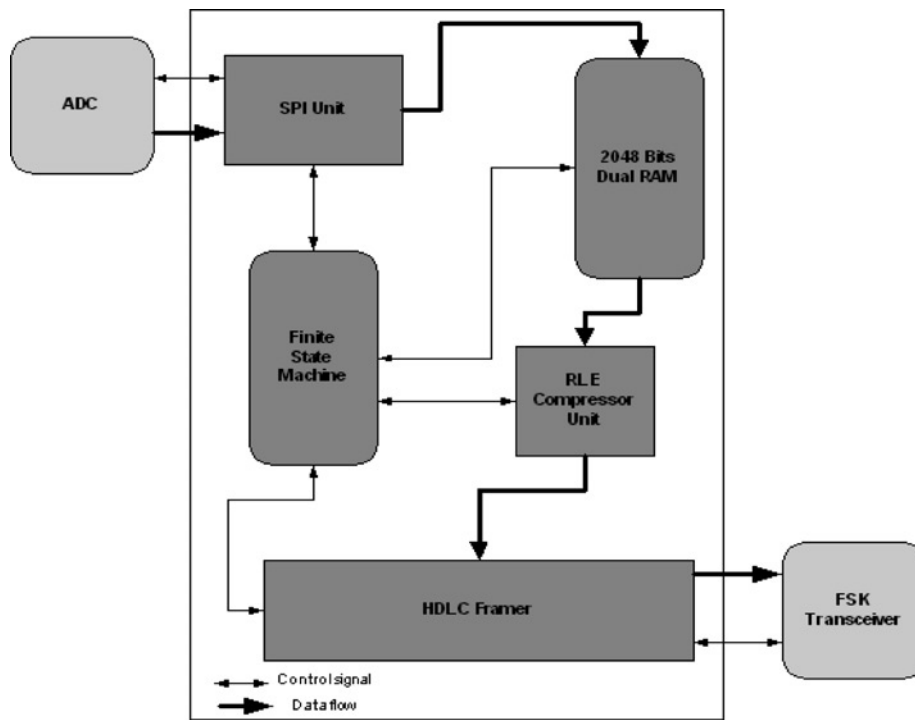


Fig. 2 Block diagram of the system



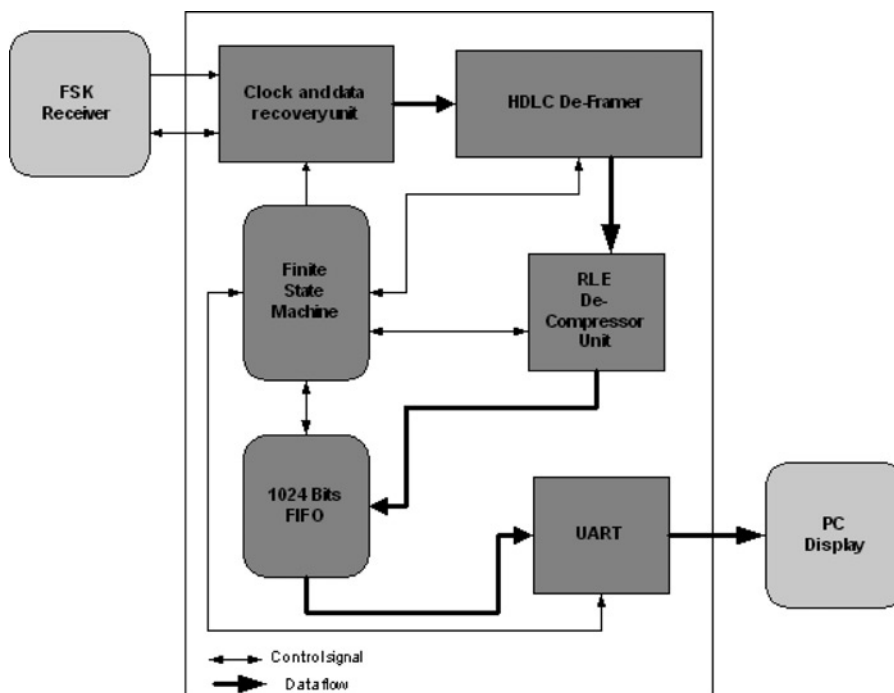
**Fig. 3** Building blocks of the transmitter FPGA

compressor units. The operation of the system units and the flow of data through the system are controlled by a main finite-state machine (FSM) controller. At the receiver end, the system units of the FPGA are organised as shown in Fig. 4. A data recovery unit is needed to extract the clock from the received bit stream. The HDLC de-framer and the RLE de-compressor blocks are designed to reconstruct the original data bytes sent by the transmitter. In this paper, Sections 6.1 and 6.2 mainly focuses on the behavioural modelling of both SPI and RLE compressor/de-compressor as they are crucial to the overall system performance.

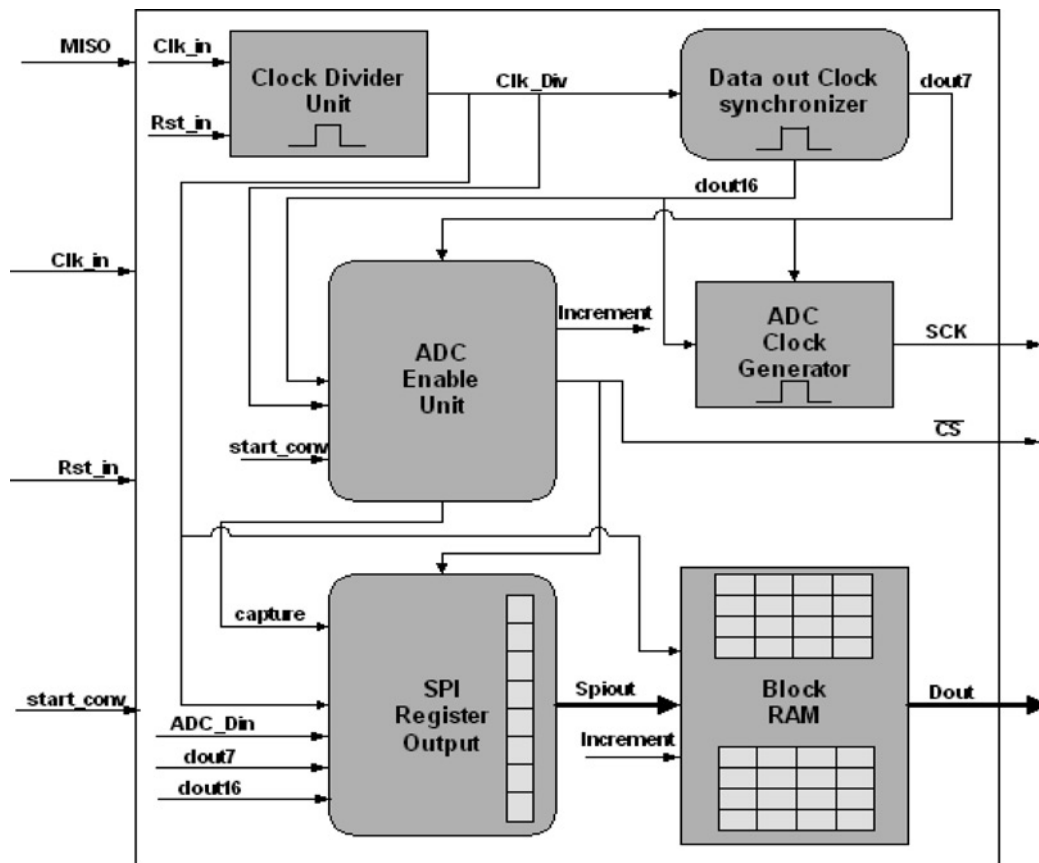
## 6.1 SPI modelling

**6.1.1 Description of the SPI protocol:** SPI is a full duplex, synchronous, serial data link. A single SPI device is configured as a master and all other SPI devices on the SPI bus are configured as slaves. The SPI bus consists of four wires:

- Serial clock (SCK): SCK is driven by the SPI master and regulates the flow of data bits. The SPI specification allows a selection of clock polarity and a choice of two



**Fig. 4** Building blocks of the receiver FPGA



**Fig. 5** Block units of the developed SPI inside the FPGA

fundamentally different clocking protocols on an eight-bit oriented data transfer.

- Master out slave in (MOSI): data output from the SPI master and input to the SPI slaves.
- Master in slave out (MISO): data input to the SPI master and output from the selected SPI slave. Only one selected slave device can drive data out from its MISO pin.

Slave chip select (CS): selects a particular slave via hardware control. Slave devices that are not selected do not interfere with SPI bus activities. The SCK, MOSI and MISO pins of all SPI devices on the SPI bus are connected together in parallel.

**6.1.2 SPI main units:** An efficient SPI unit has been modelled, as shown in Fig. 5. The MOSI signal has been omitted from the design based on the hardware requirements, where data only are transferred from the ADC to the FPGA system. The main units of the SPI are functioning as follows:

1. Clock divider unit: divides system clock by a certain factor to generate the required SPI clock frequency.
2. Data out clock synchroniser: used to generate both rising edge (dout7) and falling edge (dout16) of the ADC clock.
3. ADC enable unit: triggered on when the start\_conv signal is asserted to generate the following signals:
  - Slave CS.
  - Capture signal: to capture data transfer from the ADC to SPI register after each byte transfer.
  - Increment signal: used to change the address inside the block RAM unit.
4. SPI register unit: contains the SPI serial in/parallel out register, which is enabled when the capture signal is asserted and receives input serial data through ADC\_Din

signal. Spiout (output) signal carries the information data bytes to the distributed block RAM unit.

5. Distributed block RAM: stores the data bytes in locations determined by the increment signal.

```

//===== falling edge =====
AND3 and2instance1 (.0 (and1out),
  .I0 (inv1out),
  .I1 (fdeout1),
  .I2 (start));
INVinvinstance1 (.0 (inv1out),
  .I (QW));
FDE FDE_instance2 (.Q (fdeout2),
  .C (Clk_Div_W),
  .CE (dout7),
  .D (and1out));
defparam FDE_instance2.INIT = 1'b1;
//===== raising Edge =====
AND2 and2instance2 (.0 (and2out),
  .I0 (inv2out),
  .I1 (QW));
INVinvinstance2 (.0 (inv2out),
  .I (fdeout1));
FDE FDE_instance3 (.Q (fdeout3),
  .C (Clk_Div_W),
  .CE (dout7),
  .D (and2out));
defparam FDE_instance3.INIT = 1'b1;
//===== Multiplexer of the ADC
Chip Select=====
AND2 and2instance3 (.0 (and3out),
  .I0 (fdeout5),
  .I1 (Mod_Fjkout));
AND2 andinstance4 (.0 (and4out),
  .I0 (fdeout4),
  .I1 (inv3out));
INVinvinstance3 (.0 (inv3out),
  .I (Mod_Fjkout));
OR2 OR2_instance1 (.0 (En_Ado),
  .I0 (and4out),
  .I1 (and3out));
//===== Data Capture
FDE FDE_instance6 (.Q (fdeout6),
  .C (Clk_Div_W),
  .CE (dout7),
  .D (and2out));
defparam FDE_instance6.INIT = 1'b0;
FDE FDE_instance7 (.Q (capture),
  .C (Clk_Div_W),
  .CE (dout7),
  .D (fdeout6));
defparam FDE_instance7.INIT = 1'b0;
//===== Increment
Counter=====
FDE FDE_instance8 (.Q (fdeout8),
  .C (Clk_Div_W),
  .CE (dout16),
  .D (ModFjkout));
defparam FDE_instance8.INIT = 1'b0;
FDE FDE_instance9 (.Q (increment),
  .C (Clk_Div_W),
  .CE (dout16),
  .D (fdeout8));
defparam FDE_instance9.INIT = 1'b0;

```

**Fig. 6** Verilog-HDL code sample

**Table 1: SPI device utilisation summary using Spartan-3 S200**

Number of slices	64 out of 1920 (3%)
Number of slice flip-flops	65 out of 3840 (1%)
Number of four input look-up table (LUTs)	73 out of 3840 (1%)
Number of bonded input/output buffers (IOBs)	35 out of 173 (20%)
Number of global clock line (GCLK)	2 out of 8 (25%)
Number of digital clock manager (DCM)	1 out of 4 (25%)

**6.1.3 SPI Verilog coding and testing:** An optimised Verilog code has been used to describe the SPI block units, since it is based only on the instantiation of the basic units that can be invoked directly from the library. This is essential in the design process to remove any complexity from the model, which makes it easy to understand and debug. In addition, unnecessary resources are not added by the synthesiser tool to the code, which leads to a power-efficient model design. An example of such code for the ADC enable unit is shown in Fig. 6.

It is necessary to mention that the above code has been written using Xilinx synthesiser tool (XST), and the work is going on to make it more generic so that it could be synthesised by other tools. The summary of the synthesised SPI device utilisation is shown in Table 1, where least number of blocks has been used in the design.

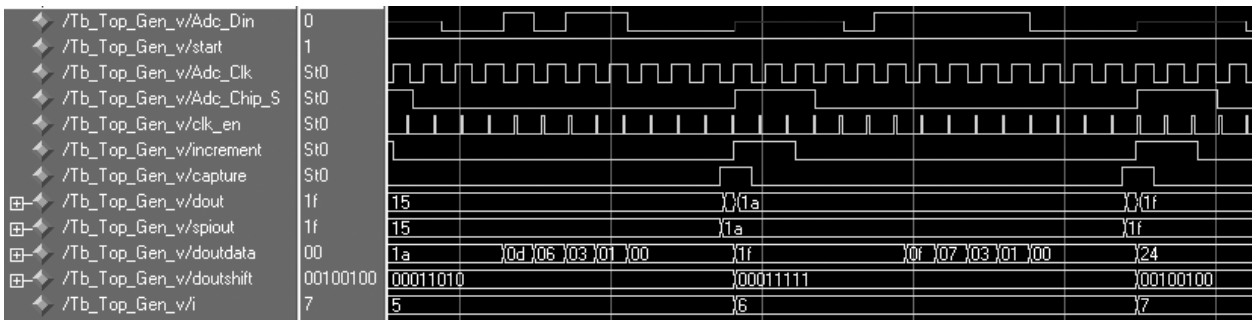
Typically, test benches have become the standard method to verify HDL designs. Test benches invoke the

functional design, and then simulate it. Accordingly, an efficient testbench has been written to mimic the behaviour of the ADC and verify the operation of the SPI system units. A sample of the output waveforms are shown in Fig. 7.

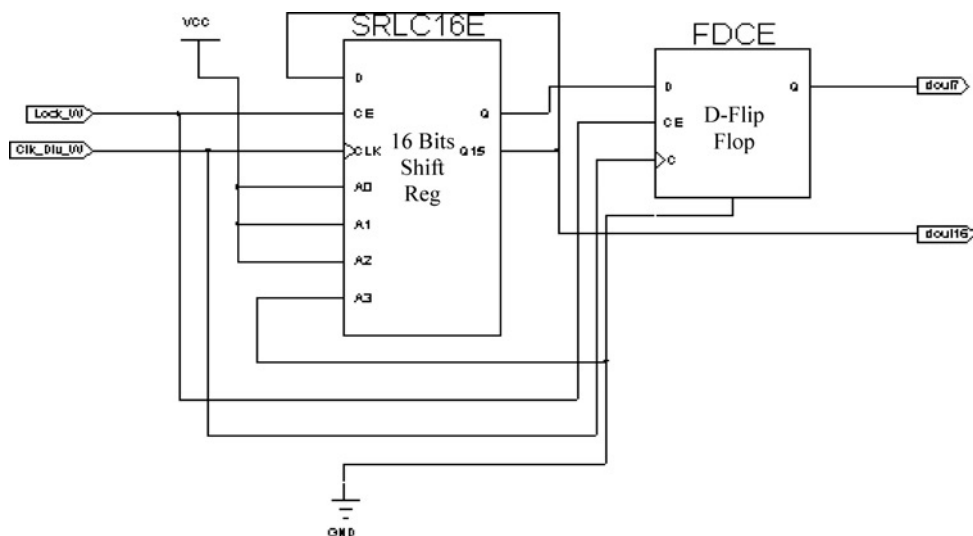
**6.1.4 SPI clocking:** Generation of the ADC clock by the SPI system was one of the difficult tasks during the system design. The requisite is that the clock rate of the ADC should be much lower than the one obtained from the DCM. A new simple way has been used to overcome this issue based on two stages. Stage 1 is presented in Fig. 8, where the DCM clock (Clk\_Div\_W) has been employed to generate two lower rate clocks, dout7 and dout16, suited for the ADC operation using 16 bits shift register. These two clocks are used later to form the correct ADC clock at the next stage. As shown in Fig. 9, simple JK flip-flop is used for this purpose to extract the ADC clock (Out\_JK). A waveforms sample for the three clocks (dout7, dout16 and Out\_JK) is shown in Fig. 10.

**6.2 RLE compressor**

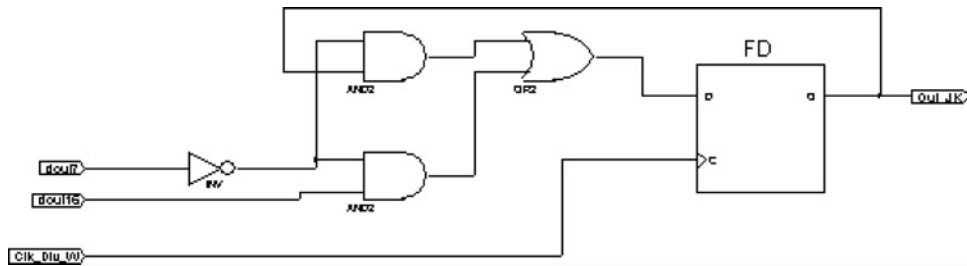
Data compression is effective when the measured signals are expected to be slow and repeatable. The task of this unit is crucial to the system power performance. RLE is a conceptually simple form of compression. RLE consists of the process of searching for repeated runs of a single symbol in an input stream, and replacing them by a single instance of the symbol and a run count. A simple flowchart of the RLE implementation is shown in Fig. 11.



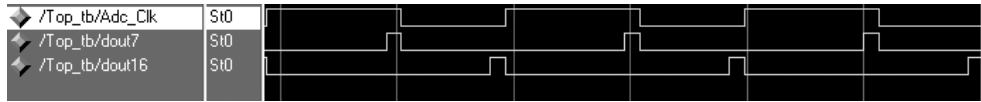
**Fig. 7** Simulation output waveforms of SPI system



**Fig. 8** SPI clock system/shift register unit



**Fig. 9** SPI clock system/JK flip-flop unit for generating ADC clock

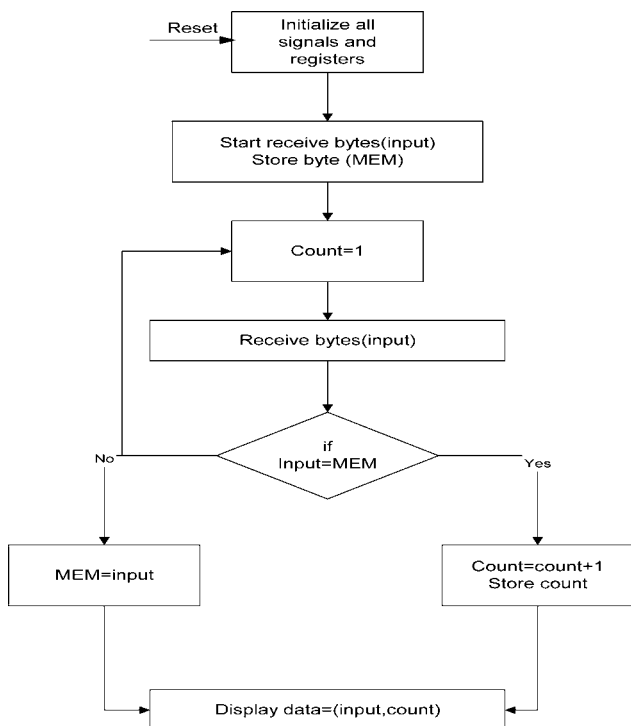


**Fig. 10** ADC clock waveforms

A sample of the output simulation waveforms is shown in Fig. 12, where the three waveforms marked are the input data bytes, valid data signal which is active (high) when there is a new or not repeated data and the two bytes (input and count) output data. The HDL code for the compressor has been written as a state machine and optimised to only two states. The code also has been synthesised successfully to achieve good device utilisation. It is worth to mention that other types of compression techniques have been considered in this work, but it was found that RLE is more suitable in terms of design complexity, power resources and compression performance.

### 6.3 RLE de-compressor

To reconstruct back the compressed data bytes, a RLE de-compressor is introduced at the receiver side FPGA. Again a simple algorithm has been adopted to implement the block tasks as shown in Fig. 13.



**Fig. 11** RLE compressor implementation

After initialisation, the de-compressor is set in a standby mode for the embedded buffer in the de-framer to be filled. Two bytes, information and count bytes are loaded into special 16-bit register each time. The count byte determines the number of times the information byte needs to be sent to the next stage.

In Fig. 14, a sample of the output waveforms obtained from running the test bench is shown and both compressed and uncompressed output data bytes have been identified.

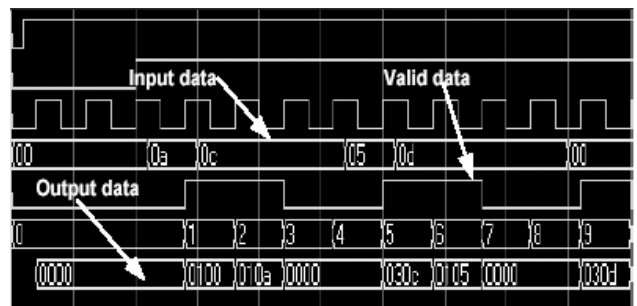
### 6.4 HDLC framer

This unit is considered to be the main core of the system model, where the data are grouped into frames and sent to the transmitter. HDLC protocol is a bit-oriented protocol that is used as a data link for most of the current communication systems [22]. The main features provided by this protocol are:

1. Synchronous operation;
2. Start and end of frame pattern generation;
3. Zero insertion and removal for transparent transmission; and
4. Cyclic redundancy check (CRC) generation for error handling.

**6.4.1 HDLC frame:** The basic structure of the HDLC frame is shown in Fig. 15.

1. Start and end flags, represented by the sequence (01111110), are required for synchronous transmission.
2. Address field is used to identify the destination address at the receiver side.



**Fig. 12** Simulation output waveforms of RLE compressor

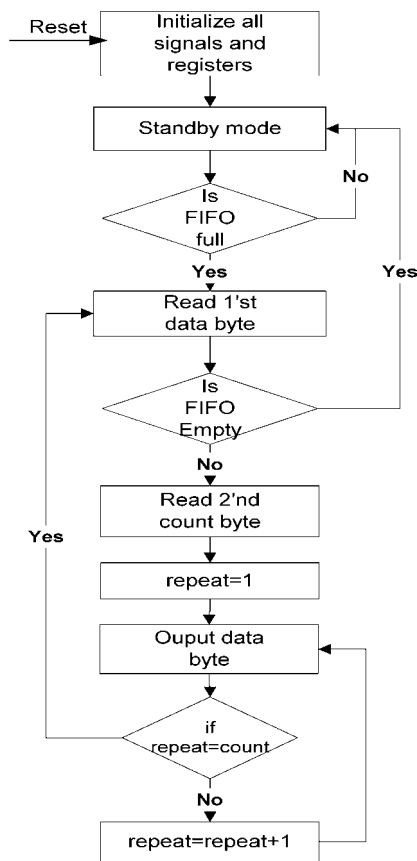


Fig. 13 Simulation output waveforms of RLE de-compressor

3. Control field is used to classify the HDLC frames according to the link configuration type.
4. Information field contains the transported data.
5. Frame check sum (FCS) is used to detect errors by adopting CRC generation.

**6.4.2 System implementation:** Recently, number of models for the HDLC protocol has been developed [23–25]. The main blocks of HDLC transmitter are shown in Fig. 16. An FSM is responsible for generating all the necessary internal control signals required by the different modules. First, the controller checks if there is a valid data output from the compressor and then starts loading the bytes into FIFO memory storage. Then the data are read serially from the memory storage and sent to the CRC module to generate the frame check sequence (FCS). The bit stuffer is responsible for examining the frame content and checking every five consecutive 1s bits

including FCS bits. If five consecutive 1s are detected, a 0-bit is inserted into the serial bit stream. This helps the receiver to distinguish the actual data transmitted. The start and end flags are generated at the final stage and attached to the frame. Also, the transmitter fills the gaps between the frames when the transmission is idle by sending a sequence of eight consecutive 1s.

The received frames are processed inversely by similar structure at the receiver control to recover the transmitted bytes. All the modules of both HDLC transmitter and receiver have been modelled, simulated and synthesised successfully.

## 7 FSK transmitter/receiver Simulink implementation

A detailed description of the simulated FSK transceiver model is given in this section.

### 7.1 FSK transmitter model

In Fig. 17, a simple FSK modulator is simulated and implemented using Simulink. A uniform random number generator is employed to output a random bit stream of 0 and 1s with a specific data rate. The input voltage to the voltage controlled oscillator (VCO) is shifted either to  $\pm 0.5$  instead of 0 or 1. This is implemented to obtain an equal frequency shift for both the tones by multiplying this input value by the sensitivity of the VCO.

The power amplifier (PA) is modelled to be a nonlinear unit, since it exhibits a higher efficiency of 60% for some of the power efficient modulation techniques as the FSK [19]; that is the FSK waveforms have no abrupt phase change and exhibit a constant envelope. Thus, the FSK signals can therefore be amplified by means of nonlinear PAs with no spectral regrowth.

### 7.2 FSK receiver model

Direct-conversion homodyne receiver is modelled as shown in Fig. 18. The low-noise amplifier (LNA) represents the first gain stage in the receiver path and its noise figure is added directly to that of the system. A small signal nonlinearity, compression, saturation, slew rate limiting and two types of noise (white and flicker) are modelled within the (LNA) unit as shown in Fig. 19.

### 7.3 Symbol (logic) detector

The design of this block has a crucial effect on the system's overall performance and its design needs to be fully understood. The main function of the unit is to recover properly

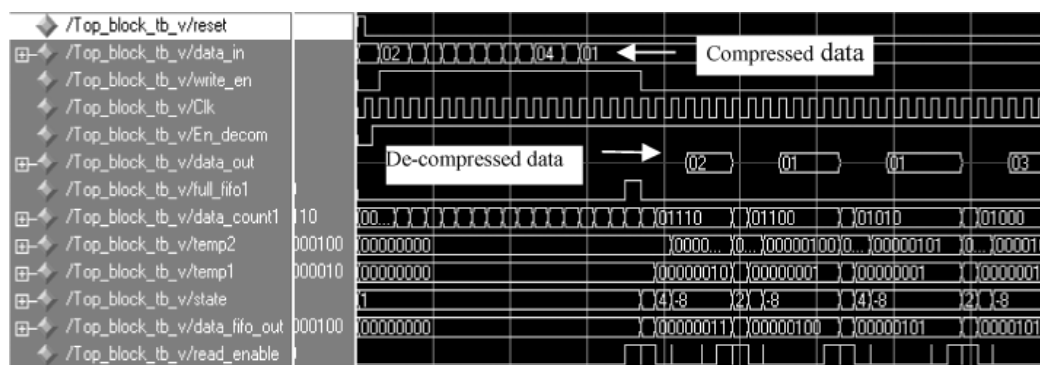
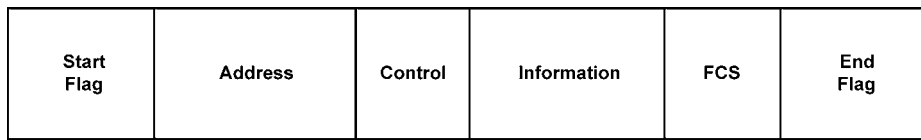
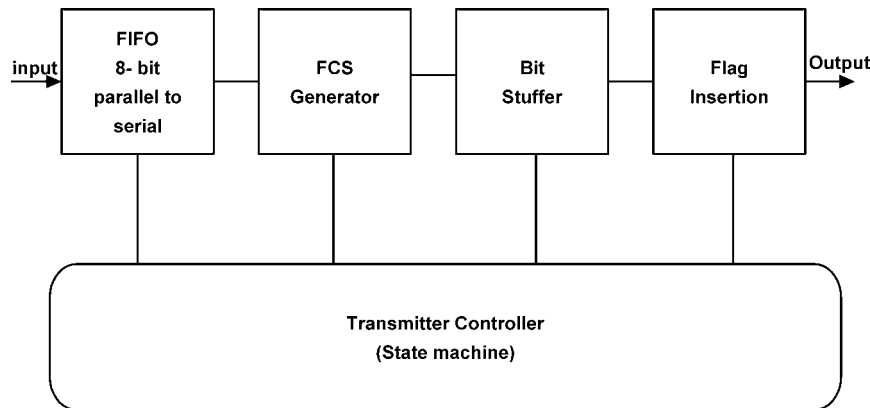


Fig. 14 Simulation output waveforms of RLE de-compressor



**Fig. 15** Main fields of the HDLC frame structure



**Fig. 16** HDLC transmitter control block diagram

the bit symbols of the input signal using a logic system. The details of this unit in Simulink are shown in Fig. 20. Combinational logic units are used to process the incoming signals from the two quadrature channels. The output bit value is extracted at the final stage by S-R flip-flop.

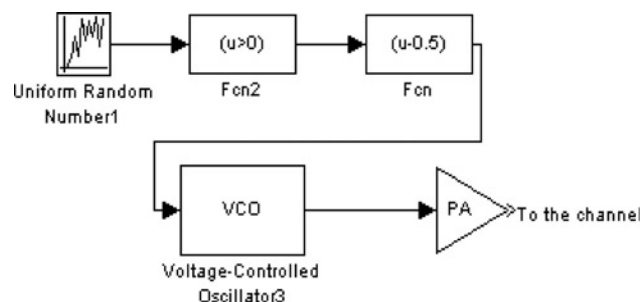
#### 7.4 Modelling channel propagation effects

The channel model is a representation of the input–output relationship of the channel in mathematical or algorithmic form. Developing the mathematical models for the propagation of the signals over a noisy transmission medium requires a good understanding of the underlying physical phenomena. One of the challenges in channel modelling is the translation of the detailed physical propagation into a form that is suitable for simulation.

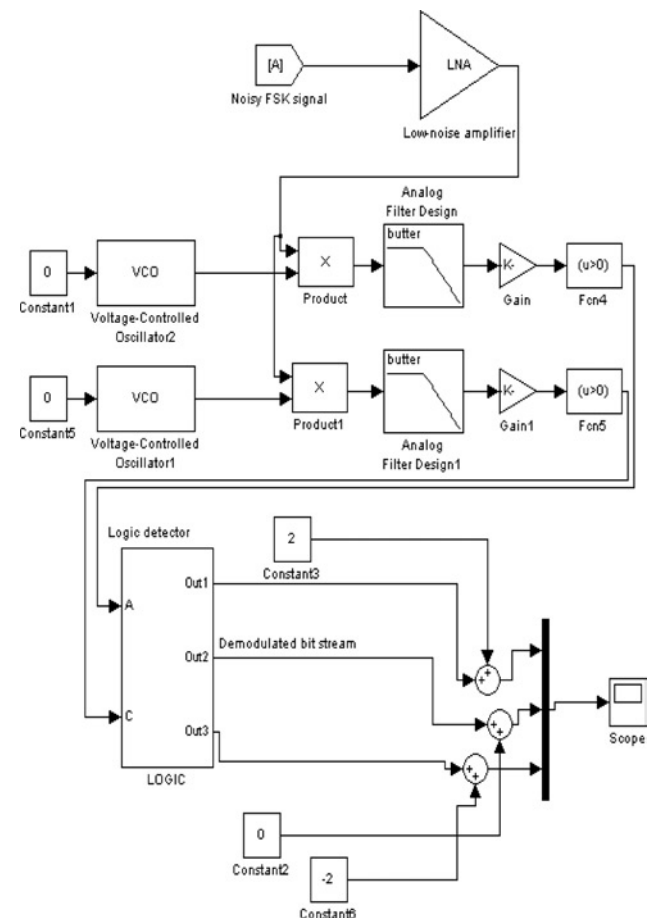
The Simulink is a powerful tool that has huge advantages in being able to build realistic mathematical models. In Fig. 21, a propagation channel model for a single-path FSK signal with the two effects, additive white Gaussian noise (AWGN) and short distance (0.5 m) path loss, is presented.

**7.4.1 Inband and outband interference model:** A major limiting factor in the performance of wireless RF systems is the interference. Sources of interferences include another transmitter device operating either in the same frequency or neighbouring frequency bands. The

two major types of system-generated interference are co-channel (inband) and adjacent (outband) channel interference. Inband channel interference is defined as undesired signals with frequency components that fall within the receiver’s RF passband. On other hand, outband channel interference is defined as signals with frequency components that are significantly removed from the receiver’s



**Fig. 17** Modelling FSK transmitter



**Fig. 18** Modelling FSK direct-conversion receiver



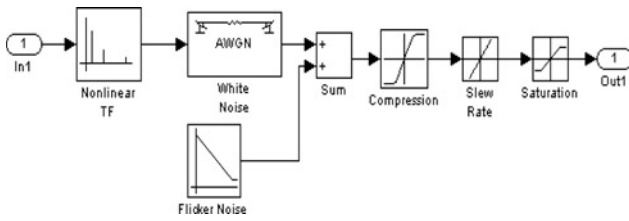


Fig. 19 Modelling a nonlinear LNA unit

RF passband. In Simulink, a simple model for system interference can be constructed by combining the output signals from a number of different FSK transmitters as shown in Fig. 22. From the figure, it is found that VCO2 and VCO3 output FSK signals with inband and outband frequencies with respect to the transmission band of VCO1.

**7.4.2 Rayleigh multipath fading model:** Rayleigh distribution is commonly used to describe time varying nature of the received envelope of a flat fading signal. Fig. 23 shows the implementation of a three-ray Rayleigh fading model using Simulink. Rayleigh generator unit is based on the rule that summation of two quadrature Gaussian noise signals obeys the Rayleigh distribution. The probability distribution function of Rayleigh distribution is given by the following equation

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & 0 \leq r < \infty \\ 0 & r < 0 \end{cases} \quad (1)$$

where  $\sigma$  is the r.m.s value and  $\sigma^2$  is the time average power of the received signal before detection. The mean value  $r_{\text{mean}}$  of the Rayleigh distribution is given by

$$r_{\text{mean}} = E(r) = \int_0^{\infty} rp(r) dr = 1.2533\sigma \quad (2)$$

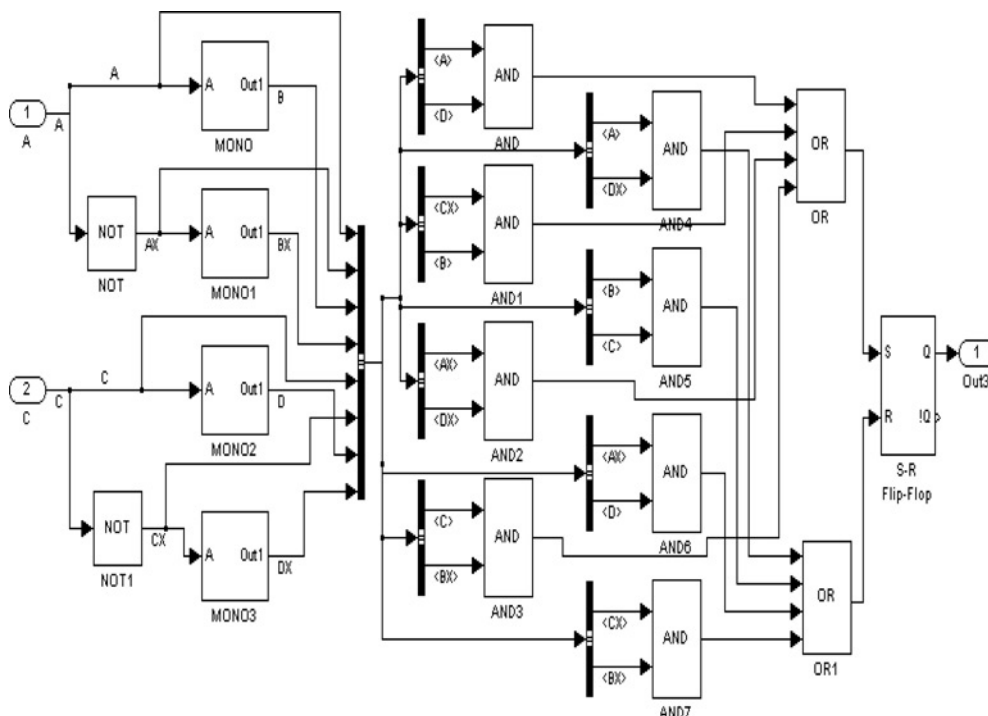


Fig. 20 Modelling symbol logic detector

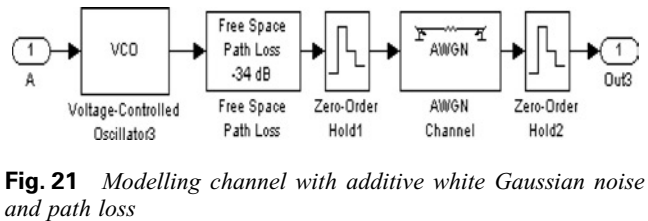


Fig. 21 Modelling channel with additive white Gaussian noise and path loss

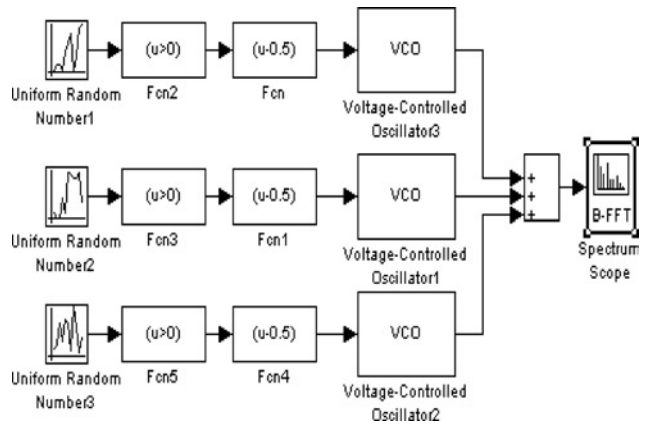


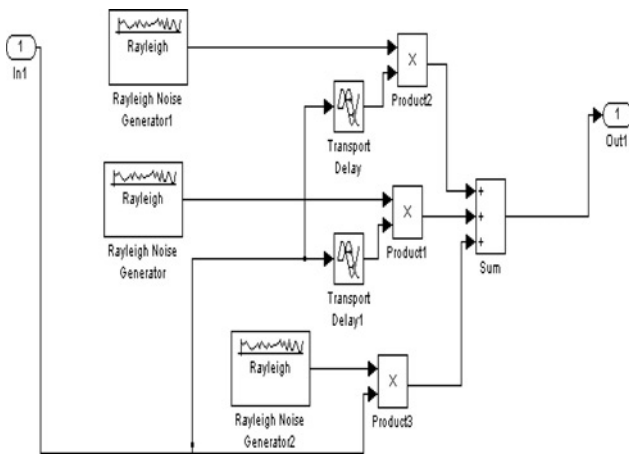
Fig. 22 Modelling the system interference

The variance of Rayleigh distribution  $\sigma_r^2$ , which represents the AC power in signal envelope is given by

$$\begin{aligned} \sigma_r^2 &= E(r^2) - E^2(r) = \int_0^{\infty} r^2 p(r) dr - \frac{\sigma^2 \pi}{2} \\ &= 0.4292\sigma^2 \end{aligned} \quad (3)$$

The median value or  $r$  is found by solving

$$\frac{1}{2} = \int_0^{r_{\text{median}}} p(r) dr \implies r_{\text{median}} = 1.177\sigma \quad (4)$$



**Fig. 23** Modelling of multipath Rayleigh fading channel

Thus, the mean and the median of Rayleigh distribution should differ only by 0.55 dB for a fading signal. The time delay units represent the multipath delay time spreads and they typically take values up to 100 ns for outdoor propagation and between 30 and 60 ns for indoor propagation [26].

## 8 Conclusions

In this paper, a mixed hardware and software simulation environments have been used to model a remote short-range wireless system. Although HDL provides many high-level abstractions and language constructions for behavioural modelling, its synthesisable subset is far too restrictive for system-level design. On the other hand, Simulink environment provides a powerful high-level mathematical modelling environment for digital communication systems that can be widely used for algorithm development and verification.

The two main operations implemented by the transmitter FPGA are compression and framing. RLE has been used to compress the stored data bytes efficiently with an optimal number of states. The HDLC protocol has been used successfully for framing the data and providing error-handling mechanism to the receiver. All the FPGA modules have been verified and implemented using Xilinx Spartan-3 device. During the development of the Verilog codes, our main goal was to make all the units synthesisable.

An FSK transmitter suitable for short-range wireless communications has been implemented using Simulink. A detailed description about both transmitter and receiver units was given. Different channel propagation effects like noise, path loss, multipath fading and interference were introduced in the model. The transceiver system performance was tested under different conditions. The implemented model showed a good capability in recovering the original data at the receiver side with different transmission frequencies. The simulation running time is considered to be the main limitation of using Simulink.

Real-time simulation of both the Simulink and the Verilog-HDL models is not the motive of this work, but to build a realistic design suitable for future hardware design. The Verilog code execution has been optimised using the code coverage feature of the ModelSim. Such feature can identify the unused 'dead' code that has a high impact on the simulation acceleration time and reducing both waste as well as risk in the targeted design.

## 9 Acknowledgments

This work was supported by the Enterprise Ireland Commercialisation Fund 2003, under technology development phase, as part of the MIAPS project, Reference no. CFTD/03/425.

## 10 References

- Folke, M., Andreasson, J., Hok, B., Linden, M., Nilsson, G., Ekstrom, M., and Backlund, Y.: 'Artefact reduction in telemetric ECG monitoring'. Proc. 12th Nordic Baltic Conf. on Biomedical Engineering and Medical Physics, Island, June 2002, vol. 2, pp. 18–22
- Reina-Tosina, J., Rao, L.M., Prado, M., and Vera, J.: 'Home health telecare and the elderly in Spain: technologies involved and methodological issues'. Proc. IEEE/EMBS 23rd Annual Int. Conf., Turkey, October 2001, vol. 4, pp. 3563–3566
- Banitsas, K.A., Istepanian, R.S.H., Tachakra, S., and Owens, T.J.: 'Modeling issues of wireless LANs for accident and emergency departments'. Proc. IEEE/EMBS 23rd Annual Int. Conf., Turkey, October 2001, vol. 4, pp. 3540–3543
- Guillen, J.M., Millet, J., and Cebrian, A.: 'Design of a prototype for dynamic electrocardiography monitoring using GSM technology'. Proc. IEEE/EMBS 23rd Annual Int. Conf., Turkey, October 2001, vol. 4, pp. 3956–3959
- Martel, S., Hatsappoulos, N., Hunter, I., Donoghue, J., Burgert, J., Malasek, J., Wiseman, C., and Dyer, R.: 'Development of a wireless brain implant: the telemetric electrode array system (TEAS) project'. Proc. IEEE/EMBS 23rd Annual Int. Conf., Turkey, October 2001, vol. 4, pp. 3594–3597
- Kim, Y., Kim, K., Shin, Y., Ahn, T., Sung, W., Choi, K., and Ha, S.: 'An integrated hardware–software co-simulation environment for heterogeneous systems prototyping'. Proc. 7th IEEE Int. Workshop Rapid System Prototyping (RSP'96), Greece, June 1996, pp. 66–77
- XILINX Documentation, System Generator User Guide, Version 8.1
- Darrel, R.J.: 'Data synchronization simulation using the mathworks'. Proc. IEEE Conf. on Communications, Converging Technologies for Tomorrow's Applications, Dallas, USA, June 1996, vol. 2, pp. 706–710
- Bateman, A., and Haines, D.M.: 'Direct conversion transceiver design for compact low-cost portable mobile radio terminals'. Proc. Conf. IEEE Vehicular Technology VTC, May 1989, pp. 57–62
- Steyaert, M., Borremans, M., Janssens, J., de Muer, B., Itoh, I., Craninckx, J., Crols, J., Morifuji, E., Momose, S., and Sansen, W.: 'A single-chip CMOS transceiver for DCS-1800 wireless communications'. IEEE ISSCC, Digest of Technical Papers, San Francisco, USA, February 1998, pp. 48–49
- Cho, T., Kang, D., Dow, S., Chun-Huat, H., and Bang, S.S.: 'A 2.4 GHz dual-mode 0.18  $\mu\text{m}$  CMOS transceiver for Bluetooth and 802.11b'. IEEE ISSCC, Digest of Technical Papers, San Francisco, USA, February 2003, pp. 88–89
- Jan, C., and Michiel, S.: 'A fully integrated 900 MHz CMOS double quadrature downconverter'. IEEE ISSCC, Digest of Technical Papers, San Francisco, USA, February 1995, pp. 136–137
- Jan, C., Kinget, P., and Michiel, S.: 'A double quadrature topology for high accuracy upconversion in CMOS transmitter'. Proc. Conf. on ESSCIRC, Neuchatel, Switzerland, September 1996, pp. 200–203
- Pui-In, M., Weng-leng, M., Seng-Pan, U., and Martins, R.P.: '1/Q imbalance modeling of quadrature wireless transceiver analog front-ends in SIMULINK'. Proc. 58th IEEE Vehicular Technology VTC, Orlando, USA, October 2003, vol. 4, pp. 2371–2374
- Gannot, G., and Ligthart, M.: 'Verilog HDL based FPGA design'. Proc. Conf. on IEEE Verilog HDL, March 1994, pp. 86–92
- Andreasson, J., Ekstrom, M., Fard, A., Castano, J.G., and Johnson, T.: 'Remote system for patient monitoring using Bluetooth/spi trade'. IEEE Sensors Proc., June 2002, vol. 1, pp. 304–307
- Shyamprakash, A., and Ravikumar, C.P.: 'Verilog modelling and simulation of a communication coprocessor for multicomputers'. Proc. Conf. on IEEE Verilog HDL, March 1995, pp. 58–66
- Blackner, A., Albl, S., and Acherr, W.: 'Configurable computing architectures for wireless and software defined radio—a FPGA prototyping experience using high level design-too-chains'. Proc. IEEE System-on-Chip Conf., Santa Clara, USA, September 2004, pp. 111–116
- Reyneri, L.M.: 'A Simulink-based hybrid codesign tool for rapid prototyping of FPGA's in signal processing systems', *Microprocess. Microsystems.*, 2003, **28**, (5–6), pp. 273–289
- Shoufeng, M., and Michael, L.: 'Mixed-signal modeling using Simulink based C'. Proc. Conf. on IEEE Behavioural and Modeling

- and Simulation (BMAS), San Jose, USA, September 2005, pp. 128–133
- 21 Razavi, B.: 'RF microelectronics' (Prentice-Hall, NJ, USA, 1998)
- 22 Michael, R., and Patel, A.: 'Performance analysis of an HDLC based protocol with data compression', *Comput. Commun. J.*, 1997, **20**, (7), pp. 567–575
- 23 Yuanlin, L., Zhigong, W., Lufeng, Q., and Bin, H.: 'Communications. Design and implementation of multi-channel high speed HDLC data processor'. Proc. IEEE Circuits and Systems and West Sino Expositions, China, June–July 2002, vol. 2, pp. 1471–1475
- 24 Bin, H., Zhigong, W., LuFeng, Q., and Yuanlin, L.: 'Design of a multi-channel high speed FIFO applied to HDLC processor based on PCI bus'. Proc. IEEE Circuits and Systems and West Sino Expositions, China, June–July 2002, vol. 2, pp. 1476–1480
- 25 Qasim, S.M., and Abbasi, S.: 'FPGA implementation of a single-channel HDLC layer-2 protocol transmitter—using VHDL'. Proc. 15th Int. Conf. on Microelectronics ICM, Cairo, Egypt, December 2003, pp. 265–268
- 26 Anderson, J., Rappaport, T.S., and Yoshida, S.: 'Propagation measurements and models for wireless communications channels', *IEEE Trans. Commun. Mag.*, 1995, **33**, (1), pp. 42–49