A Differential QPSK Modem Using the TMS320C6711 DSK

Stephan Weiss, Matthew R. Bennett, Edward Gibson, and Neil C. Tisdale

School of Electronics & Computer Science
University of Southampton, SO17 1BJ, UK
Tel. +44(0)2380-597645, Fax +44(0)2380-594508
{sw1,mrb04r,nct04r}@ecs.soton.ac.uk

ABSTRACT

We report on a student project implementing a D-QPSK modem on a TI C6711 DSK. The modem incorporates functionalities such as QPSK symbol generation, differential encoding and decoding, transmit- and receive filtering, quadrature modulation, timing synchronisation, and bit error detection. Both transmitter and receiver have to be operated concurrently and in real time. The DSK's on-board digital to analogue and analogue to digital converters were employed to interface the transmitted and received signals of the modem. Based on a set of specifications, the various solutions implemented by the students had to be compatible and be able to synchronise with each other.

1. INTRODUCTION

As part of a four year degree at the University of Southampton, students in electronic engineering can attend modules in communications, which are either entirely taught or contain course work in terms of simulations that train advanced concepts but often omit basic but necessary communications problems such as synchronisation and timing recovery. The fourth year module "real-time systems design" was therefore aimed at implementing a simple modem which combines the teaching of real-time system issues such as scheduling with essential knowledge of practical communications. Similar work has been the topic in past fourth year modules [**?**], but were targetted at smaller student numbers, where a whole group would build a single modem together.

Due to a larger number (12) and the availability of a DSP lab equipped with C6711 DSKs, each student had to implement their own modem based on a set of specifications, which should be able to communicate with any other student's realisation via the DSK's audio I/O facility. The modem was to transmit a pseudo-random bit sequence at 2kb/s encoded as differential QPSK (D-QPSK) and modulated onto a carrier of 2kHz. This rather low specification was imposed by the codec's 8kHz sampling rate, but realistic transmit and receive filtering operations as well as timing synchronisation in the receiver demanded careful design and consideration of real-time system aspects when dealing with the multirate nature of the overall system.

The course consisted of a taught component on applied signal processing, as well as some programming experience with interrupts and I/O functionality via the DSK's codec. The latter was provided through an initial lab session, from where on the students were free to select their own implementation. In the following, we first review the specifications of the assignment in Sec. 2. and review one of the solutions.

2. SYSTEM SPECIFICATION AND OVERVIEW

The overall required functionality of the modem is shown in Fig. 1, with independent units for transmitter (Tx) and receiver (Rx) running concurrently on the same device. On the Tx side, a pseudo-random number (PN) generated bitstream at 2 kb/s is mapped onto a differentially encoded QPSK symbol stream at 1 baud. The later is required in order to blindly recover the signal in the receiver despite the rotational invariance of the QPSK constellation pattern [1]. The symbol stream is upsampled to 8kHz and filtered by a given root raised cosine filter for pulse shaping spanning 10 symbol periods in order to suppress inter-symbol interference in the receiver. The resulting signal is quadrature amplitude modulated onto a carrier of 2 kHz and output via the DSP serial port connected to a codec running at 8kHz sampling rate.

The receiver acquires an analogue signal via the DSK's analogue to digital converter at 8kHz. This signal is demodulated to baseband, receive filtered by a second 80 coefficient long root raised cosine filter, and decimated by a factor of 8 to the correct baud rate. In the decimation process one of the eight polyphase components of the signal provides an optimum reception and minimum bit error ratio (BER) in the presence of channel noise. To identify which of the 8 polyphase components to retain, an early-late gate scheme is suggested to determine the accurate phase shift of the sampling grid and to allow the tracking of potential clock frequency mismatches between transmitter and receiver [1].

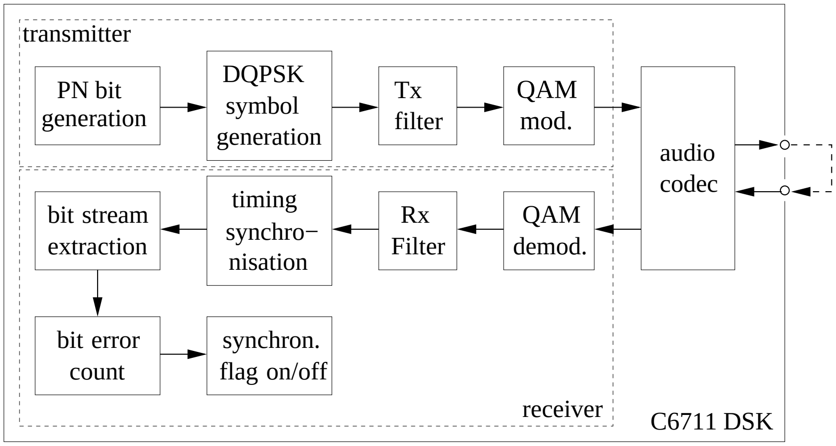

Fig. 1. A 2kb/s modem running concurrent independent transmitter and receiver units, with pulse shaping, modulation, timing synchronisation and sequence detection.

In order to determine whether the correct sequence was detected, the transmitter's PN sequence generator polynomial was assumed to be known. Using the on-board LEDs, the transmitter should switch on an LED if the correct sequence was detected, and switch it off if synchronisation was lost or detection errors occurred.

3. SYSTEM COMPONENTS

The setup given in Fig. 1 underlines the modularity of the specified system, whereby dual functions in transmitter and receiver, such bit generation and error count, QPSK symbol generation and bit stream extraction, transmit and receive filtering, and modulation and demodulation can be designed and tested together. The following subsections provide details on the modem functionality, to which the students were exposed in the taught component of the course.

3.1. Bit Generation and Synchronisation

Ideally we would like to transmit a user-defined bit sequence. However, in many communications systems (such as the UMTS mode UTRA TDD) a pilot sequence is embedded in the transmitted signal which will allow the receiver to synchronise with the transmitter. Thus, here we only consider the transmission of such a pilot sequence, which is to be produced by a pseudo-random number (PN) generator.

A 16-bit linear feedback shift register, implementing a maximum length sequence which only repeats every $2^{16}-1$ iterations, can implement the PN generator according to the flow graph in Fig. 2, whereby the input to the shift register is drawn by XORing the state values in registers 4, 13, 15 and 16. In each iteration, the right most state value shifted out off the register forms the current bit of the bit stream to be transmitted.

The correctness of the bit stream in the receiver can be assessed by the system in Fig. 3, which fills the states of the same 16 bit shift register, comparing the feedback value with the most recent bit. Once in steady state, the output synchronisation flag is permanently high in the absence of bit errors. The number of zeros occuring then represents the number ofencountered bit error.

3.2. QPSK Symbol Generation and Bit Extraction

QPSK maps two bits onto a complex symbol that can take on four possible values in the constellation plane. Multiplexing the bitstream into inphase and quadrature signals $I[n]$ and $Q[n]$ with $I[n],\ Q[n] \in \{0;1\}$, whereby n is the time index of the QPSK signal $x_0[n]$,

$$x_0[n] = \sqrt{2}(\frac{1}{2} - I[n]) + j\sqrt{2}(\frac{1}{2} - Q[n]) = e^{j\psi[n]} \quad (1)$$

running at a half of the bit rate, i.e. 1kHz. This is cointained in the two left-most blocks of Fig. 4.

The original bit stream can be extracted from the reconstructed QPSK sequence in the receiver, $\hat{x}_0[n]$, by taking real and imaginary parts, and multiplexing the recovered in-

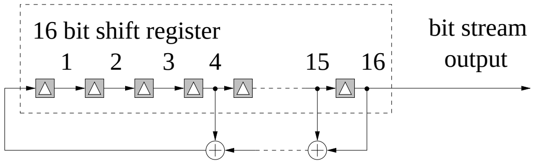

Fig. 2. Maximum length sequence based pseudo-random number generator.

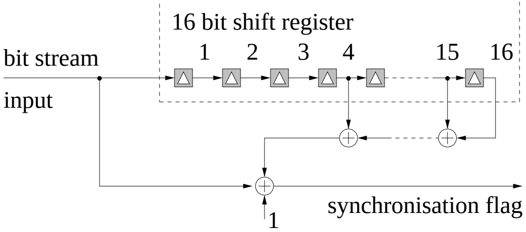

Fig. 3. The linear feedback shift register of Fig. 2 as a synchronisation tool.

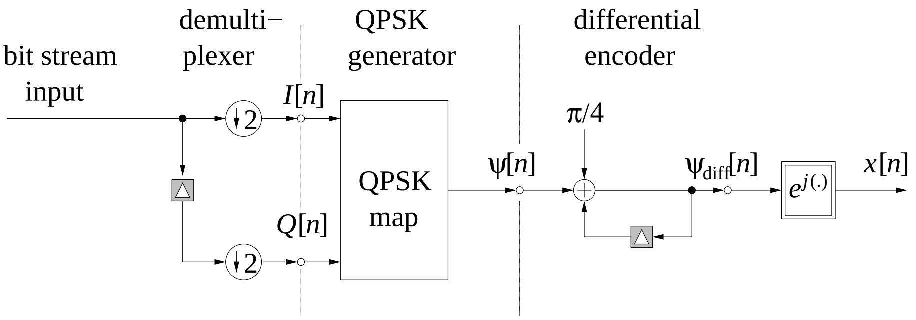

Fig. 4. (left) differential QPSK symbol generation.

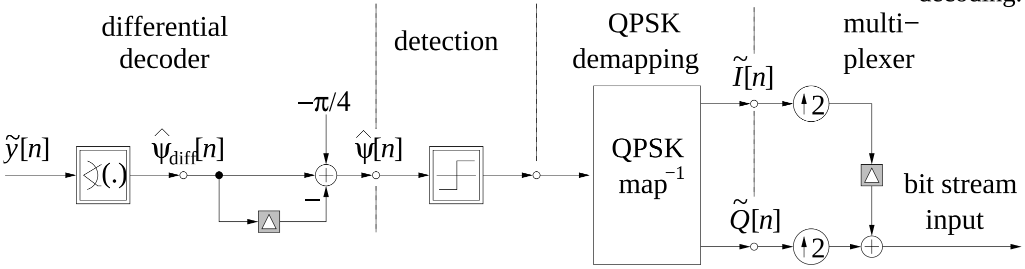

Fig. 5. (left below) differential QPSK decoding.

phase and quadrature components

$$\hat{I}[n] = \frac{1}{2} - \frac{1}{2}\text{sign}\{\Re\{\hat{x}_0[n]\}\} \tag{2}$$

$$\hat{Q}[n] = \frac{1}{2} - \frac{1}{2}\text{sign}\{\Im\{\hat{x}_0[n]\}\} \quad , \tag{3}$$

where sign$\{\cdot\}$ is the signum function.

3.3. Differential QPSK

Since the QPSK constellation is invariant to shifts by $\pi/2$, π, and $3\pi/2$, the receiver suffers from an ambiguity. To mitigate this, differential encoding is applied to the phase of the transmitted QPSK symbols. This can be based on the phase of the QPSK signal $x_0[n] = e^{j\psi[n]}$,

$$\psi_{\text{diff}}[n] = \psi_{\text{diff}}[n-1] + \psi[n] + \frac{\pi}{4} \tag{4}$$

using the circuit in Fig. 4. Alternatively, rather adding phase values, the QPSK symbols can be multiplied,

$$x[n] = x[n-1]\ \hat{x}_0[n]\ e^{j\pi/4} \tag{5}$$

whereby $x[n] = e^{j\psi_{\text{diff}}}$ is the differentially encoded QPSK (DQPSK) symbol stream.

Decoding can be based on the phase values of the received DQPSK signal $\hat{x}[n]$ as shown in Fig. 5 according to

$$\hat{\psi}[n] = \hat{\psi}_{\text{diff}}[n] - \hat{\psi}_{\text{diff}}[n-1] - \pi/4 \tag{6}$$

with $\hat{y}[n] = e^{j\hat{\psi}_{\text{diff}}[n]}$ or alternatively

$$\hat{y}_0[n] = \hat{y}[n]\ \hat{y}^*[n-1]\ e^{-j\pi/4} \tag{7}$$

3.4. Transmit and Receive Filtering

In the transmitter we need to oversample the symbol stream by a factor of $N = 8$ in order to obtain a signal to be sent to the DAC, by inserting $N-1$ zeros in between every original DQPSK sample. This is performed by the 8-fold expander shown in Fig. 6. The resulting signal is broadband, and needs to be band-limited through interpolation by a suitable transmit filter — here a root-raised cosine filter with 81 coefficients and impulse response $h[m]$ as characterised in Fig. 7. Together with the receive filter in the receiver, the transmit filter forms a Nyquist system, i.e. imposes no intersymbol interference (ISI) on the transmitted symbol stream.

Note that the input to the real valued filter is a complex valued signal. Filtering can be performed such that the real part of the output $\hat{x}[m]$ is calculated by convolving the real part of the input with the impulse response, while the imaginary part of the output is obtained by convolving the imaginary part of the input with $h[m]$. Thus, filtering is applied to the real and imaginary part of the input separately.

An efficient implementation of the Tx filter would make sure that inside the filter multiplications with expanding zeros of the input signal are avoided.

A receive filter with a root raised cosine characteristic as

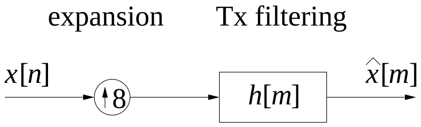

Fig. 6. Transmit filtering with root raised cosine filter $h[m]$ after expansion by factor $N = 8$.

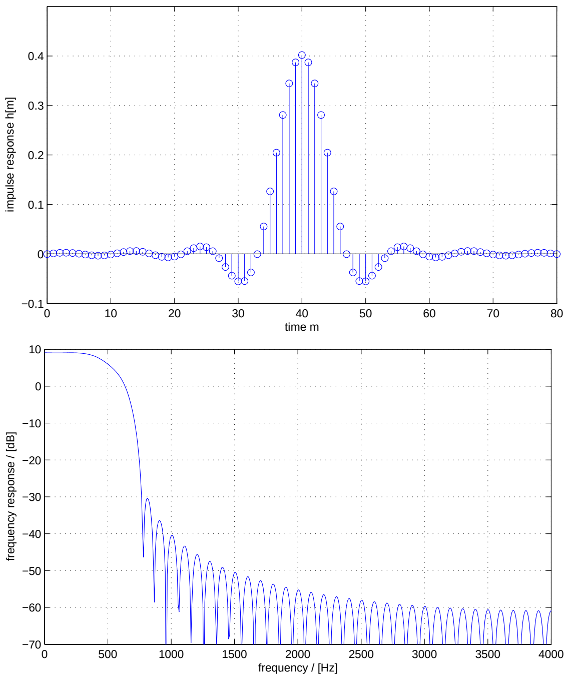

Fig. 7. Tx filter characteristic: impulse response (left) and frequency response (right).

shown in Fig. 7 is employed to cancel ISI and potentially suppress out-of-band noise in the receiver. The filtering operation is depicted in Fig. 8.

Note that the filter is followed by a decimation stage by $N = 8$, i.e. only every 8th sample in $y[m]$ will be kept. Firstly, for the filtering operation this means redundancy, as most of the outputs that, which have been tediously calculated, will be discarded. Secondly, there is an ambiguity, as there are $N = 8$ possibilities, depending on which samples of $y[m]$ will be kept. To select the correct sampling instance is the aim of the synchronisation unit.

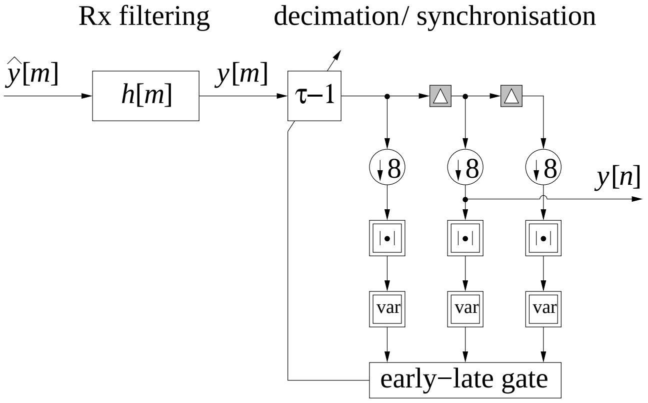

Fig. 8. Receive filtering with root raised cosine filter $h[m]$ and decimation by $N = 8$ after timing synchronisation.

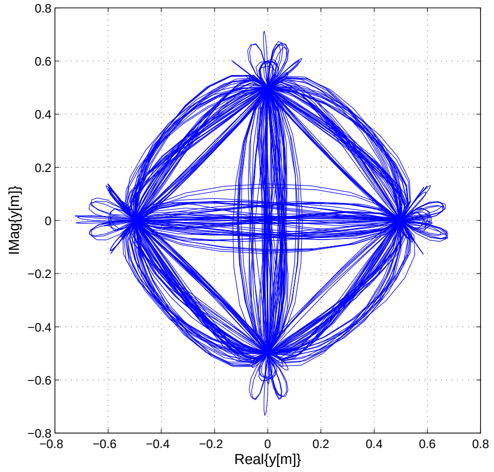

Fig. 9. Real vs imaginary part of $y[m]$.

3.5. Modulation and Demodulation

To transmit the data, quadrature amplitude modulation of the QPSK symbol stream is employed. This can be implemented as

$$\begin{aligned} s[m] \quad = \quad & \Re\{\hat{x}[m]\}\cdot\cos(2\pi f_c/f_s\; m) \;\; + \\ & + \;\Im\{\hat{x}[m]\}\cdot\sin(2\pi f_c/f_s\; m) \end{aligned} \tag{8}$$

whereby a carrier frequency $f_c = 2000$ Hz and a sampling rate of $f_s = 8000$ Hz has to be utilised.

The signal $r[m]$ received by the codec is demodulated as

$$\hat{y}[m] = r[m]\cdot e^{j2\pi f_c/f_s\; m} \quad . \tag{9}$$

This creates a baseband version of the transmitted signal, and a modulated version oscillating at twice the carrier frequency, $2f_c$. Usually, a lowpass filter would be applied to $\hat{y}[m]$ in order to isolate the baseband signal. Here, this lowpass operation is inherently performed by the Rx filter in Sec. 3.4.

3.6. Timing Synchronisation

The most complex operation in the modem is the timing synchronisation in the receiver, at which stage a decision has to made on the exact sampling instance to retrieve the DQPSK symbols $\hat{y}[n]$ at 1kbaud from the 8 kHz receive filtered signal in Fig. 8. The timing synchronisation algorithm selected here is an early-late gate method [1].

Considering a noise free case, the signal $y[m]$ after receive filtering will follow the trajectories displayed in Fig. 9. Displaying real and imaginary part separately in eye diagrams over two symbo periods in Fig. 10(top), it is evident that for one sampling instance out of 8 possibilities, the trajectory passes through a constellation point of constant magnitude. Therefore, observing the statistics of the magnitude

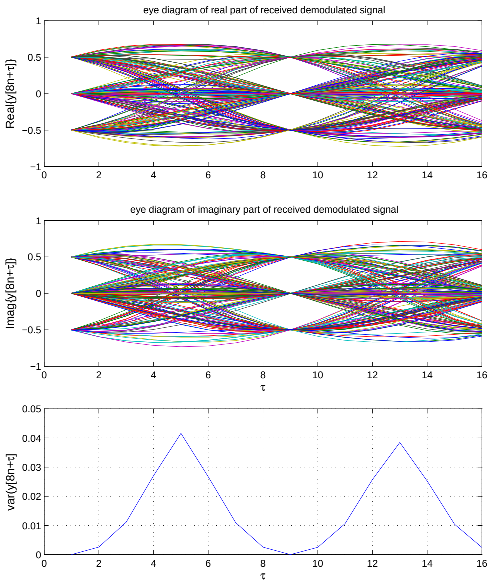

Fig. 10. Eye diagrams for real (inphase, top) and imaginary (quadrature, middle) component of the signal $y[m]$, which is wrapped-around in intervals of two symbol periods; (bottom) variance of $|y[m]|$ at the different sampling points.

for various sampling instances, we note that the variance of the sample modulus is only zero if the correct sampling instance is chosen. Fig. 10(bottom) providing those variance indicates that the variance is monotonous and a gradient approach can be selected to attain the sampling instance with minimum variance. In the presence of channel noise, the entire curve in Fig. 10(bottom) would translated in height, but otherwise remain unchanged.

To obtain the gradient, Fig. 8

4. REAL TIME ASPECTS

For the implementation on the TMS320-C6711 DSK, the students had to carefully consider the various sampling rates in the system. In the solution considered here, the system was governed by hardware interrupts triggered by the McBSP hardwired to the DSK's codec, which occurred at a rate of 8 kHz. With double buffering of the received and transmitted data, which was operated in the Tx and Rx interrupt service routines (ISR), the transmitter functions were performed every 8th sampling period, triggered by a low priority software interrupt. Similarly for the receiver functions, on the 8th Rx interrupt a software interrupt was issued to perform the receiver functions. This organisation ensured concurrency of Rx and Tx and avoided an unevenly balanced amount of processing to be performed in each ISR.

In order to efficiently implement the system, the Rx and Tx filters were implemented in polyphase form [2], operating any filtering at the lowest possible sampling rate and avoiding multiplications with zeros in the upsampling stage of the transmitter as well as the calculation of discarded filter outputs in the decimation stage of the receiver. Further efficiency was exploited by noting that real and imaginary part of the carrier signal are alternatingly zero. For the receiver, the timing synchronisation was based on statistics collected over sections of 500 successive symbols, which was verified to yield sufficiently confident estimates by software simulation.

For testing, the modularity of the system in Fig. 1 was exploited. Firstly the PN sequence generator was tested together with the detector. Secondly DQPSK modulation and demodulation were included. In a further step, Tx and Rx filtering were introduced, followed by QAM modulation and demodulation. The timing recovery could be added, with the Tx directly feeding into the Rx block in software. Finally, the DAC/ADC system could be included. Testing was aided by a good off-line debugging of source code, and the use of DIP switches and LEDs on the board. The later was necessary, since the real time aspect rendered the use of other debugging tools, e.g. the use of printf() statements, difficult. Finally, execution graphs in Code Composer Studio proved to be very useful for determining the correct real-time functionality of the implementation.

5. TESTING AND VERIFICATION

6. CONCLUSIONS

A student assignment on producing an exemplary real time system in form of a DQPSK modem has been described, and one solution implemented by a student has been briefly outlined. Of the 11 implemented systems, 8 were successful in communicating according to the specifications, as demonstrated by compatibility with the systems of other student. The full paper will provide details of the exercise and exemplary implementation, and also hint at approaches chosen by other students for their implementation as well as compare the various systems in terms of their real time capabilities. Recommendation for similar future assignments will be made.

For further information, please visit `www.ecs.soton.ac.uk/˜sw1/`

7. REFERENCES

[1] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice Hall, Englewood Cliffs, 1988.

[2] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, 1993.