

# Quad-level Bit-Stream Signal Processing on FPGAs

Chiu-Wa Ng, Ngai Wong, Hayden Kwok-Hay So, and Tung-Sang Ng  
Department of Electrical and Electronic Engineering  
University of Hong Kong  
E-mail: {cwng, nwong, hso, tsng}@eee.hku.hk

## Abstract

*Quad-level bit-stream signal processing (BSSP) circuits are implemented and their performances are compared with previously published tri-level and bi-level BSSP implementations on FPGAs. BSSP refers to the process of performing computation directly on over-sampled delta-sigma modulated signals to eliminate the need of resource consuming decimators and interpolators. Quad-level BSSP offers better performance than their bi- and tri-level counterparts at the expense of higher resource utilization. Using a digital phase locked loop (DPLL) and a quadrature phase-shift keying (QPSK) demodulator as application examples, the effectiveness of quad-level BSSP on FPGAs is studied. The BSSP approach will be contrasted with conventional multi-bit implementations using built-in digital signal processing blocks in modern FPGAs.*

## 1. Introduction

In a system that utilizes bit-stream signal processing (BSSP) technique, digital signal processing (DSP) is performed using the over-sampled bit-stream output from the sigma-delta modulator (SDM) directly without the use of decimators and interpolators. The hardware complexity of BSSP systems is potentially much lower than that of conventional multi-bit processing systems, giving rise to a low-cost and resource-efficient way of signal processing [1-6].

One way to increase the signal-to-noise-and-distortion ratio (SNDR) of the bit-stream is to utilize a multi-level quantizer. Quantizing the analog input into two, three, or four levels results in bi-, tri-, and quad-level BSSP processing elements with increasing hardware complexity. Such increase in resource utilization counteracts the benefit of elimination of decimators and interpolators of BSSP.

At the same time, modern FPGAs, including many low cost variants, contain highly efficient multi-bit multipliers or DSP blocks as hard macros. Unlike full custom application specific integrated circuit (ASIC) designs in which every transistor used will contribute to the overall silicon resource consumption, the use of such embedded DSP blocks in FPGAs does not consume any general logic resources, giving rise to a new dimension in system resource consumption trade-offs.

The goal of this paper is therefore to study the overall system tradeoffs when such quad-level BSSP blocks are utilized on FPGAs, taking into account the presence of embedded multi-bit DSP blocks. In particular, two moderately sized applications – a digital phase lock loop (DPLL) and a quadrature phase-shift keying (QPSK) demodulator introduced in [1] – are implemented using efficient quad-level BSSP blocks on an FPGA. The architectures of each sub-module will be presented. Resource utilizations of their FPGA implementations and their signal-to-noise performance are also contrasted against conventional binary and tri-level realizations. Furthermore, resource utilizations of conventional multi-bit implementations of the two applications are also estimated on an FPGA with built-in high-performance DSP blocks, which serves to provide insights for system designers to understand design tradeoffs for employing quad-level BSSP on modern FPGAs.

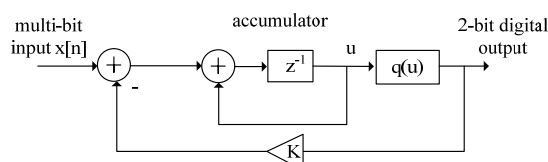
## 2. Quad-level BSSP Circuit Modules

### 2.1. Digital Sigma-Delta Modulator (DSDM)

A DSDM is shown in Fig. 1. It converts a multi-bit input  $X[n]$  into a quad-level bit-stream output  $y[n]$ . In the diagram, the feedback gain  $K$  must match with the dynamic range of the  $X[n]$ . The

quantizer function  $q(u)$  has the following characteristic:

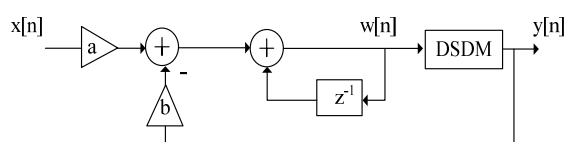
$$q(u) = \begin{cases} 3, & 2K \leq u \\ 2, & 0 \leq u < 2K \\ 1, & -2K \leq u < 0 \\ 0, & u < -2K \end{cases}$$



**Figure 1: Digital sigma-delta modulator (DSDM).**

## 2.2. Bit-Stream Lowpass Filter (LPF)

A first-order bit-stream LPF [7] is depicted in Fig. 2.



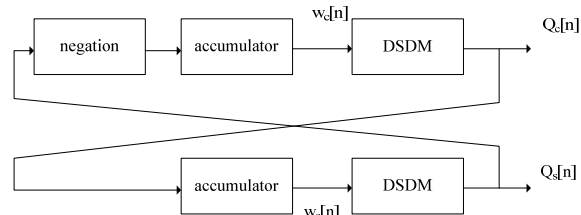
**Figure 2: Bit-stream lowpass filter (LPF).**

As both the input  $x[n]$  and output  $y[n]$  are 2-bit bit-streams, the two gain blocks in Fig. 2 can be implemented by multiplexers. To demonstrate the performance gain of the quad-level bit-stream LPF over the bi- and tri-level counterparts, simulation of the bi-, tri- and quad-level filters is carried out. These filters have a normalized cut-off frequency about 0.00186. The SNDR of the quad-level design is 65.2 dB while that of the tri-level and bi-level LPF are 62.5 dB and 53.6 dB, respectively. The SNDR is determined by the ratio of the output power of a sinusoid, at a normalized frequency of 0.00189 and with a unity amplitude, to the total noise power in the frequency band of interest. The over-sampling ratio (OSR) is 128.

## 2.3. Bit-Stream Numerically Controlled Oscillator (NCO)

To construct a bit-stream numerically controlled oscillator (NCO), a fixed frequency sigma-delta based oscillator will first be presented. As shown in Fig. 3, this oscillator consists of three types of modules: (i) two quad-level DSDMs with feedback gain  $K$ ; (ii) two accumulators with upper and

lower limits of  $\pm A$ , where  $1 \ll A < 3K$ ; and (iii) a module for the negation operation. The increment values of the accumulator are -3, -1, 1 or 3 when the quad-level input is 0, 1, 2 or 3, respectively. With such configuration, the two outputs of the oscillator, ( $Q_c[n]$  and  $Q_s[n]$ ), will be two sigma-delta modulated sinusoids with a phase difference of  $\pi/2$  at the frequency of  $1/(2\pi K)$ .

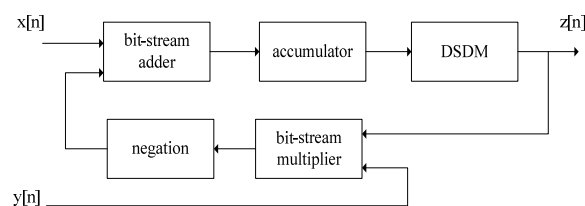


**Figure 3: Sigma-delta based oscillator.**

To construct an NCO, the feedback gain  $K$  of the two DSDMs is changed by  $\Delta K$  from the center value  $K_0$  by a bit-stream control signal  $c[n]$ , according to [1]. Simulation shows that the average SNDR of the quad-level NCO is 49.4 dB while that of the tri-level and bi-level design are 45.3 dB and 41.6 dB, respectively. The parameters used in the simulation are as follows:  $A = 75$ ,  $K_0 = 79$ ,  $\Delta K = 3$ , OSR = 128.

## 2.4. Bit-Stream Divider

The block diagram of a bit-stream divider is shown in Fig. 4. Let  $\bar{x}$  denotes the average value of  $x[n]$ . The average output  $\bar{z}$  of the quad-level bit-stream divider converges to  $\bar{x}/\bar{y}$ .



**Figure 4: Bit-stream divider.**

## 2.5. Bits-Stream Square Root Circuit (SQRT)

The architecture of the quad-level SQRT is shown in Fig. 5. The average output  $\bar{z}$  of the quad-level bit-stream SQRT converges to the square root of  $\bar{x}$ .

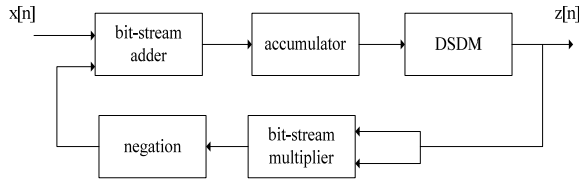


Figure 5: Bit-stream square root circuit.

### 3. Application Examples

In this section, two application examples, namely, a DPLL and a QPSK demodulator are described and the FPGA implementation results of the bi-, tri- and quad-level designs are presented for comparison. The circuits are implemented with the Xilinx Virtex-5 XC5VLX30 FPGA using the design tool ISE 9.1i.

#### 3.1 DPLL

A Type-1 DPLL [8] is shown in Fig. 6. In this particular implementation, the normalized input frequency is  $1/512$ .  $A$ ,  $K_0$  and  $\Delta K$  are set to 80, 81 and 3, respectively. The OSR is 128. Simulations confirm that all the bit-stream DPLLs can synchronize to the input signal at steady state. The SNDRs of the bi-, tri- and quad-level DPLL outputs are 39.1 dB, 48.6 dB and 56.9 dB respectively. Table 1 shows the FPGA implementation results of the 3 DPLL designs.

Table 1: Implementation Results of Bi-level and Quad-level DPLL Designs.

	Bi-level	Tri-level	Quad-level
No. of LUTs	122	208	517
No. of FFs	79	91	383

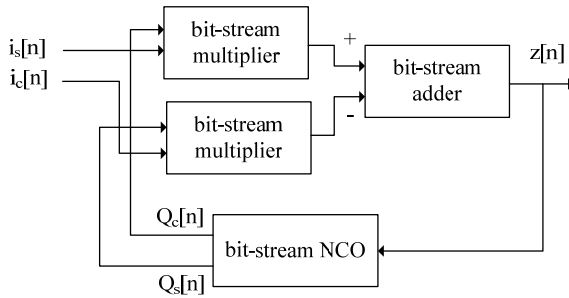


Figure 6: Type-1 DPLL.

#### 3.2 QPSK Demodulator

The QPSK demodulator in [1] has been implemented using the proposed quad-level signal processing building blocks. The QPSK

demodulator consists of the synchronization part and the phase detection part as depicted in [1]. The specification of this particular implementation for the bi-, tri- and quad-level designs is shown in Tables 2 & 3. In Fig. 7, the quad-level design achieves more well-defined constellation which leads to a better performance. The FPGA implementation results of the three bit-stream QPSK demodulators are shown in Table 4. For a comparison on the hardware complexity of the bi-, tri and quad-level BSSP circuit modules, Table 5 shows the FPGA resource utilization of individual component in this particular QPSK demodulator realization.

Table 2: Specification of the QPSK Demodulator.

Item	Specification
Input carrier	Sigma-delta modulated sinusoidal wave with a normalized frequency of 0.002
Phase shift interval	5000
NCO parameter	$A = 75$ , $K_0 = 79$ , $\Delta K = 4$ for bi- and tri-level; $\Delta K = 3$ for quad-level

Table 3: Specification of the Bit-Stream LPFs.

Bit-stream LPF	Bi-level		Tri-level		Quad-level	
	Cut-off frequency	Gain	Cut-off frequency	Gain	Cut-off frequency	Gain
(C)	$6.22 \times 10^{-4}$	1.5	$6.22 \times 10^{-4}$	2	$6.22 \times 10^{-4}$	2
(S)	$6.22 \times 10^{-4}$	1.5	$6.22 \times 10^{-4}$	2	$6.22 \times 10^{-4}$	2
(L)	$6.22 \times 10^{-3}$	4	$6.22 \times 10^{-3}$	4	$6.22 \times 10^{-3}$	4
(R)	$3.11 \times 10^{-4}$	3	$3.11 \times 10^{-4}$	2	$3.11 \times 10^{-4}$	2
(X)	$3.11 \times 10^{-4}$	1	$3.11 \times 10^{-4}$	1	$3.11 \times 10^{-4}$	1
(Y)	$3.11 \times 10^{-4}$	1	$3.11 \times 10^{-4}$	1	$3.11 \times 10^{-4}$	1

Table 4: Implementation Results of Bi-level and Quad-level QPSK Demodulator Designs.

	Bi-level	Tri-level	Quad-level
No. of LUTs	539	840	2120
No. of FFs	383	420	1591

Table 5: Implementation Results of BSSP Circuit Modules.

Component	Bi-level		Tri-level		Quad-level	
	No. of FFs	No. of LUTs	No. of FFs	No. of LUTs	No. of FFs	No. of LUTs
DSDM	11	11	11	13	11	11
LPF	21	30	20	30	22	32
NCO	36	63	36	74	40	94
Divider	41	51	47	85	191	248
SQRT	38	47	41	73	185	248

### 4. Discussion

In this paper, we have presented various quad-level BSSP circuit modules which are extended from the conventional 1-bit designs. We have also compared the signal-to-noise performance and

resources requirement of these components with existing bi-level and tri-level designs. In general, the quad-level implementations achieve better signal-to-noise performance than their bi-level and tri-level counterparts at the expense of higher circuit complexity. Due to the higher complexity of the quad-level bit-stream multiplier, for applications that require multiplier, much more FPGA resources (LUTs and FFs) are required in the quad-level case as shown in Table 1, 4 and 5. Thus for the performance and complexity tradeoff, it seems that tri-level BSSP is the best amongst the three. Tri-level BSSP achieves significantly better signal-to-noise than bi-level BSSP with moderate increase in circuit complexity as compared with quad-level BSSP.

Comparing the BSSP approach with the conventional Nyquist rate approach targeted for FPGA implementation, the incorporation of DSP48E slices in Virtex-5 series allows very efficient multi-bit implementation of signal processing circuits. For example, when all the multipliers and accumulators are fitted into DSP48E elements, an equivalent eight-bit implementation of the Type-1 DPLL described in Section 3.1 only consumes 64 LUTs plus 8 DSP48E slices. It seems that the BSSP approach is not as resource-efficient as the conventional multi-bit approach when implemented in FPGA using the “free” DSP resources.

On the other hand, one of the advantages in BSSP is that the decimator and interpolator for the conventional Nyquist approach are not required. Depending on applications, the hardware resources for a decimator can be as low as 556 LUTs as in [9] or 2116 Virtex-4 slices as in [10]. As one decimator is required for each analog input and one interpolator is required for each analog output, the total amount of FPGA resources for decimator and interpolator can be large. Thus depending on the complexity of the final system, the BSSP approach can still be more resource-efficient than the conventional multi-bit approach when the number

of LUTs in implementing the BSSP circuits is smaller than or comparable to that in the decimator and interpolator implementation for the Nyquist rate multi-bit approach.

## 5. References

- [1] H. Fujisaka, R. Kurata, M. Sakamoto, and M. Morisue, “Bit-stream signal processing and its application to communication systems,” *IEE Proceedings – Circuits, Devices and Systems*, 149, (3), pp. 159-166, 2002.
- [2] P. O’Leary and F. Maloberti, “Bit stream adder for oversampling coded data,” *Electronics Letters*, 26, (20), pp. 1708-1709, 1990.
- [3] C. W. Ng, N. Wong and T. S. Ng, “Bit-stream adders and multipliers for tri-level sigma-delta modulators,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 12, pp. 1082-1086, Dec 2007.
- [4] C. W. Ng, N. Wong and T. S. Ng, “Tri-level bit-stream signal processing circuits and applications,” in *Proceedings of International Conference on Signal Processing and Communications Systems (ICSPCS 2007)*, paper ID 92, Dec 2007.
- [5] C. W. Ng, N. Wong, and T.S. Ng, “Quad-level Bit-stream Adders and Multipliers with Efficient FPGA Implementation,” *Electronics Letters*, vol. 44, (12), pp.722-724, 2008.
- [6] P. W. Wong, “Fully sigma-delta modulation encoded FIR filters,” *IEEE Transactions on Signal Processing*, vol. 40, (6), pp. 1605–1610, June 1992.
- [7] D. A. Johns, and D. M. Lewis, “Design and analysis of delta-sigma based IIR filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, (4), pp. 233-240, 1993.
- [8] F. M. Gardner, *Phaselock Techniques*. Hoboken, NJ: Wiley-Interscience, 2005.
- [9] L. Fucjik, A. S. Kuncheva, T. Mougel, and R. Vrba, “New VHDL Design of Decimation Filter for Sigma-Delta Modulator,” *Proceedings of 2005 Asian Conference on Sensors and the International Conference on new Techniques in Pharmaceutical and Biomedical Research*, pp. 204-207, 2005.
- [10] N. Khouja, K. Grati, and A. Ghazel, “Low Power FPGA-Based Implementation of Decimating Filters for Multistandard Receiver,” *Proceedings of International Conference on Design and Test of Integrated Systems on Nanoscale Technology*, pp. 10-14, 2006.

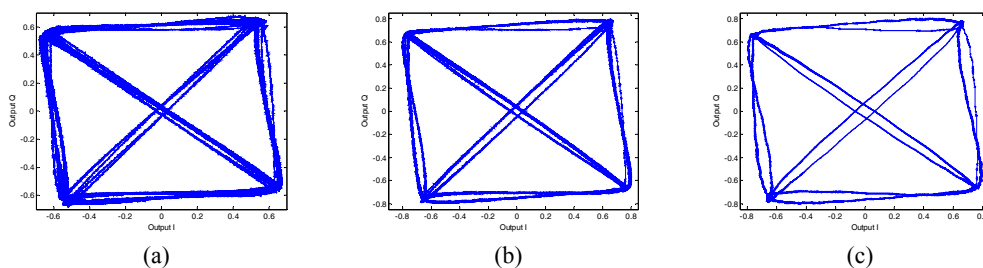


Figure 7: Output constellation plots: a) bi-level design; b) tri-level design; c) quad-level design.