

# ON THE DESIGN AND FPGA IMPLEMENTATION OF REAL-TIME SCANNED-ARRAY 2D FREQUENCY-PLANAR BEAM FILTERS

Arjuna Madanayake and Len Bruton, *IEEE*

Department of Electrical and Computer Engineering, University of Calgary, Alberta, Canada

Phone: 1 403 220 4881 , fax: 1 403 282 6855, email: bruton@ucalgary.ca  
web: http://www-mddsp.enel.ucalgary.ca

## ABSTRACT

We propose a field programmable gate array (FPGA) circuit-based 2D IIR single-chip spatio-temporal filter having low hardware complexity for the purpose of processing 2D scanned-array sampled signals. The filter is suitable for the selective real-time filtering of broadband spatio-temporal plane waves by employing low-order highly-selective 2D frequency-planar beam-shaped filter pass-bands. The design and hardware co-simulation is described for a Xilinx FPGA chip. Employing a linear array of  $N$  sensors, temporal sampling rates up to  $f_{AS}/N$  MHz per sensor are feasible using one A/D converter and a FPGA chip, clocked at  $f_{AS}$  MHz. A directional audio example is given for CD quality propagating sound waves.

## 1. INTRODUCTION

The potential applications of real-time 2D frequency-planar (FP) filters include, but are not limited to, such fields as directional audio imaging, sonar, seismic and ultrasound imaging and intermediate frequency (IF) beam forming. In particular, it has been shown that practical-BIBO stable infinite impulse response (IIR) FP 2D digital filters are useful for the selective filtering [1][2] or jamming [3] of 2D broadband spatio-temporal plane waves (PWs) based on their directions of arrival (DOAs).

While the synthesis and design of the transfer functions of such filters are well documented in the literature [1][2], real-time *single-chip* VLSI circuit implementations have not, to our knowledge, been achieved. This lack of practical progress is largely due to the extensive data acquisition hardware associated with the temporal sampling and A/D conversion at each of the multiple sensors, the extensive hardware challenges associated with the on-chip acquisition of the data from the sensors and the lack of a suitable single-chip real-time 2D IIR filter circuit. In this contribution, all three of these limitations are addressed using single-chip FPGA technology.

The proposed method yields a simple low-cost low-order VLSI-CMOS real-time 2D frequency-planar (beam-shaped) IIR filter having low hardware complexity, low device resource consumption and low power consumption. It employs only one time-division multiplexed (TDM) A/D converter for data acquisition from the multiple sensors. In section 2, the sampling method is described. In section 3, a proposed 2D IIR filter architecture is described that is suitable for processing 2D input signals obtained using the described scanning method, including a FPGA device single-chip implementation. Our method lends itself to implementation using corresponding VLSI CMOS application specific integrated circuits (ASICs).

## 2. ASYNCHRONOUSLY-SAMPLED 2D SIGNALS

### 2.1 Synchronously-Sampled 2D Spatio-Temporal Signals

Consider the continuous-domain spatio-temporal 2D input signal  $w(x, t)$  and the corresponding to the 2D Fourier transform pair  $w(x, t) \Leftrightarrow W(\omega_x, \omega_t)$ . Then the conventional synchronously-sampled version of the input is given by  $w(n_1\Delta x, n_2\Delta T) \Leftrightarrow W_{SYNC}(\omega_x, \omega_t)$ , where  $0 \leq n_1 < N_1$  corresponds to the linear array of  $N_1$  sensors, uniformly spaced  $\Delta x$  apart and sampled every  $\Delta T$  seconds. At each instant of time  $n_2\Delta T$ , all  $N_1$  sensor signals are *synchronously sampled* and then digitized by  $N_1$  analog-to-digital (A/D) converters. For highly-selective 2D filters, such as the narrow 2D beam and fan filters used for PW filtering, the spatially-long 2D impulse response  $h(n_1\Delta x, n_2\Delta T)$  of the transfer function typically requires that  $N_1$  exceed 100, implying over 100 synchronously-operating A/D converters and therefore significant hardware complexity.

### 2.2 Asynchronously-Sampled 2D Spatio-temporal Signals

Asynchronous sampling strategies may be employed in order to allow the time-division multiplexing (TDM) of A/D converters among the linear array of  $N_1$  sensors and thereby to reduce the number of A/D converters, albeit at the expense of higher A/D temporal sample rates. For single-chip single-A/D 2D filter implementations, we propose the following straightforward asynchronous A/D TDM strategy and we consider its effect on the design and implementation of single-chip 2D filters in a real-time FPGA environment. Uniformly scanning the sensor array yields a 1D pipelined temporal signal given by

$$w_{SCAN}(k\Delta T_{AS}) = w(n_1\Delta x, (n_2 - \frac{n_1}{N_1})\Delta T), \quad 0 \leq n_1 < N_1, \quad 0 \leq n_2$$

$$k = n_2N_1 + n_1 \quad (1)$$

so that only one A/D converter and an analog multiplexer (MUX) are required to asynchronously sample all  $N_1$  sensors in the linear

spatial array. The individual samples are acquired at time instants  $(n_2 - n_1 / N_1)\Delta T$  so that the time between all *spatio-temporally adjacent* samples is  $\Delta T_{AS} = \Delta T / N_1$ . We need only one (albeit high-speed) time-multiplexed A/D converter having the uniform A/D sample time  $\Delta T_{AS}$ .

The 2D Fourier spectrum  $W_{SCAN}(\omega_x, \omega_t)$  of  $w(n_1\Delta x, (n_2 - (n_1/N_1))\Delta T)$  in equation (1) is typically only slightly different than  $W_{SYNC}(\omega_x, \omega_t)$ . From equation (1), the 2D scanned samples are at locations in  $(x, t)$  given by  $\mathbf{A}[n_1\Delta x \quad n_2\Delta T]^T$  where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ -1/N_1 & 1 \end{bmatrix} \quad (2)$$

It then follows by the linear transformation property that

$$W_{SCAN}(\boldsymbol{\omega}) = W_{SYNC}(\mathbf{A}^{-T}\boldsymbol{\omega}) / \det \mathbf{A} = W_{SYNC}(\mathbf{A}^{-T}\boldsymbol{\omega}) \quad (3)$$

where  $\boldsymbol{\omega} \equiv [\omega_x \quad \omega_t]^T$ , implying that the scanning operation modifies the 2D input spectrum, relative to the synchronous sampling case, by the transformation  $\mathbf{A}^{-T}$  which, for  $N_1 \gg 1$ , is approximately equivalent to a small rotation of the 2D input spectrum, given by  $\theta = \tan^{-1}(1/N_1)$  for the normalized case  $\Delta T \equiv \Delta x \equiv 1$ . If the required angular selectivity of the 2D pass-band of the filter is much larger than  $\theta$ , the effect of this transformation may be neglected. Otherwise, it must be taken into account by *rotating the transfer function of the conventional synchronously-sampled filter by  $\theta$  to allow for the effects of asynchronously-sampled scanning*.

### 3. A SINGLE-CHIP FPGA 2D FIRST-ORDER IIR FREQUENCY-PLANAR FILTER

We now describe a method by which practical-BIBO stable low-order 2D IIR recursive filters may be implemented for the processing of scanned-array 2D spatio-temporal signals. The computable first-order normalized 2D IIR input-output difference equation is completely described by

$$y(n_1, n_2) = \sum_{i=0}^1 \sum_{j=0}^1 a_{ij} w(n_1 - i, n_2 - j) - \sum_{i=0}^1 \sum_{j=0}^1 b_{ij} y(n_1 - i, n_2 - j) \quad (4)$$

$i=j \neq 0$

over all  $n_{1,2}$  such that  $0 \leq n_1 \leq (N_1 - 1)$  and  $0 \leq n_2$  where  $a_{ij}, b_{ij}$  are feed-forward and feed-back filter coefficients, with the spatio-temporal zero initial conditions (ZICs) given by  $w(-1, n_2) \equiv y(-1, n_2) \equiv 0$  and  $w(n_1, -1) \equiv y(n_1, -1) \equiv 0$ .

The architecture of the proposed algorithmically-equivalent pipelined 1D hardware implementation of this difference equation is summarized in the signal flow graph (SFG) of Figure 1 and corresponds to Algorithm 1 in Appendix 1. The corresponding FPGA hardware uses a vector processor to compute equation (4) at each asynchronous clock sample period. The array-scanning method allows the spatial delay operations in equation (1) to be converted to equivalent time delays in Figure 1 using digital delay lines [4][5]. Algorithm 1 and the equivalent hardware in Figure 1 test for the ZICs using a hardware version of the function `SDP_Block_FunctionX()` which is mapped on to hardware as the

spatial delay processor (SDP) circuit. The SDP circuits in the branches of the SFG handle the ZICs by connecting a register-stored 0 value to the output of the SDP when the ZICs are required and allowing the signal to feed through otherwise.

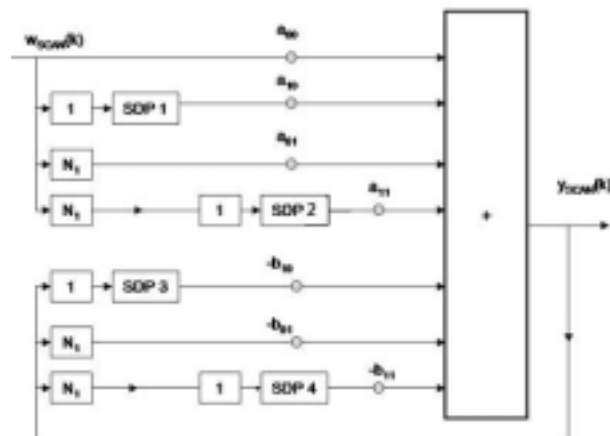


Fig.1 Hardware SFG of Proposed Architecture.

The required 1D temporal output signal  $y(N_1, n_2\Delta T)$  is obtained from the 1D temporal output in Figure 1, using

$$y(N_1, n_2\Delta T) = y_{SCAN}((n_2 N_1 + N_1 - 1)\Delta T_{AS}) \quad (5)$$

A down sample by  $N_1$  circuit which is in turn fed via a FIFO buffer makes up the hardware mapping of equation (5) to produce the 1D output signal  $y(N_1, n_2\Delta T)$ . The FIFO hardware consists of a clocked delay line using register flip-flops locked at  $f_{AS} = 1/\Delta T_{AS}$  Hz.

#### The Xilinx Vertex II FPGA Single-Chip xc2v2000 Implementation:

The FPGA circuit is designed to selectively filter sampled broadband 2D spatio-temporal PWs having a DOA at 30 degrees to the temporal axis in the normalized  $(n_1, n_2)$  space-time domain, a CD quality temporal sampling frequency of 44.1 kHz and a FP pass-band having -3 dB angular frequency selectivity of 2 degrees at a temporal frequency of 20 kHz. We use  $N_1 = 100$  in order achieve an adequate coverage of the spatial transient response and the coefficients of the 2D filter are chosen from [1] to achieve these specifications. Such a filter is useful for broadband audio beam filtering applications where high-selectivity of the DOA is required.

#### 3.1 Hardware Mapping Of 2D FP Filter To A Xilinx Vertex II FPGA Device Using The Xilinx System Generator (XSG) Design Tool

The input is obtained using a  $N_1 : 1$  analog multiplexer and an A/D converter. The multiplexer circuit has a digital input selection port that is used to generate the multiplexed asynchronous sampled 1D input signal  $w_{SCAN}(k\Delta T_{AS})$ . The digital control port is addressed in hardware as described in algorithm 2 of the Appendix. The analog output of the  $N_1 : 1$  multiplexer is sampled by the A/D converter at the asynchronous sample frequency  $f_{AS}$ .

All feed-forward and feed-back branches in the SFG of Figure 1 are concurrently implemented on the FPGA circuit. From Algorithm

1, we show that a signal component shifted by a single spatial location is equivalent to a temporal delay of  $\Delta T_{AS}$  for the scanned signals  $w_{SCAN}(k\Delta T_{AS})$  or  $y_{SCAN}(k\Delta T_{AS})$ . Also, temporal sample delays  $\Delta T$  are obtained by delaying a scanned signal component by  $\Delta T_{AS}N_1$ . These clocked delay elements are realized in hardware as FIFO buffers using serially-cascaded registers of length 1 and  $N_1$  respectively, which are clocked at a rate  $f_{AS}$ .

### 3.2 Xilinx Implementations of SDP and SFG Using Parallel FPGA Hardware.

The SDP blocks are implemented in hardware using a resettable up counter that stores the variable SDPn1, to be incremented at each asynchronous sample period and reset when  $SDPn1=N_1$ . (i.e. this counter is reset when the spatial computation has ended and the last sensor has been sampled). The FPGA implementation using the XSG Design Tool is shown in Figure (2). The SFG, based on the SDP implementation, is mapped to a parallel vector processor as shown in Figure (5). This real-time implementation produces a multiplexed output sample for every asynchronous clock sample.

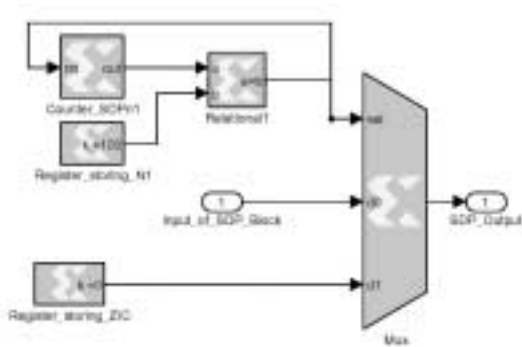


Fig. 2 The XSG Circuit Using XSG Blocks

The XSG Design Tool produces a design that is targeted towards a Xilinx XtremeDSP Kit, consisting of a Nallatech® DIME II BenOne® carrier board, a Xilinx Vertex II® xc2v2000 FPGA chip and a Nallatech Ben-Adda® daughter board with in-built A/D converters. The integrity of the 2D FP filter has been confirmed with bit and cycle accurate modelling of the XSG design within the Simulink environment. The further inclusion of a “XtremeDSP Compilation” XSG block leads to the co-simulation library, which allows the filter to operate in “stepped mode” on the Xilinx xc2v2000 chip. Communication with Matlab/Simulink on the host PC is via the USB interface. The Simulink XSG co-simulations of a 16 bit 2D IIR FP filter on the XtremeDSP Kit is shown in Figure 3. Co-simulation has been used to verify the normalized 2D unit impulse response  $h(n_1, n_2)$  of the hardware in the spatio-temporal domain and confirmation of the 2D FP frequency response has been obtained as the magnitude of the 2D FFT of  $h(n_1, n_2)$  as shown in Figure 4. Note the exceptional frequency domain selectivity achieved by this low-order single-chip filter.

## 4. CONCLUSIONS

A hardware efficient novel FPGA single-chip pipelined 2D first-order IIR filter architecture is proposed where the spatial array is scanned using a single on-chip TDM A/D converter. Implementation as a FP filter is described using the XSG Design Tool and has been tested on the Xilinx XtremeDSP Kit. Co-simulation confirms the capability for real-time performance of single-chip narrow 2D FP filters. We report here the hardware co-simulation of a narrow FP beam broadband design for a linear array of  $N_1 = 100$  microphones, each sampled at the CD-quality audio temporal sampling frequency  $f_s = f_{as} / N_1 = 44.1kHz$ , requiring an easily achievable FPGA clock frequency of only 4.41 MHz. The Xilinx FPGA chip can operate at clock frequencies in excess of 100 MHz when XSG IP Cores are used. Higher speeds of operation may indeed be possible with better optimized designs of the constituent building blocks.

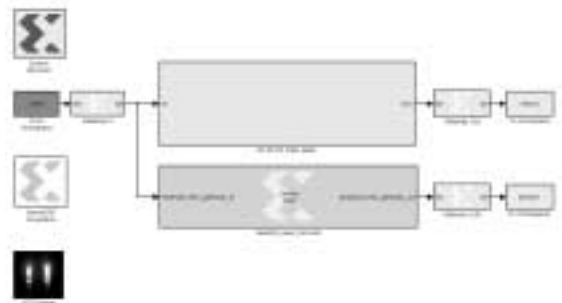


Fig. 3 Xilinx Hardware Co-Simulation Model Targeted At The Xilinx XtremeDSP Kit.

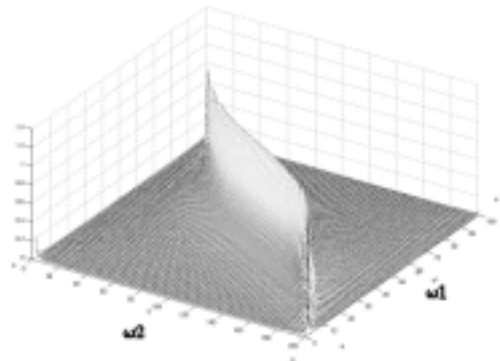


Fig. 4 Magnitude 2D FFT Of The 2D Unit Impulse Response Via Hardware Co-Simulation.

The proposed 2D Xilinx FPGA design may be used as a building block for creating real-time 2D IIR fan filter banks, which will operate at up to 12 MHz temporal sampling frequencies per sensor, on Xilinx Vertex II FPGA chips, when 25 spatial samples are used in an asynchronous scanning sampling structure. Future on-chip VLSI ASIC implementations are thus deemed to be feasible based on these FPGA prototype circuits. This method is directly extendable to

3D IIR filters and especially to 3D cone filter banks for 3D plane wave filtering.

## REFERENCES

- [1] L.T. Bruton and N.R. Bartley, "Three-Dimensional Image Processing Using The Concept of Network Resonance", IEEE Transactions on Circuits and Systems, Vol. CAS -32, No. 7, July 1985.
- [2] L.T. Bruton, "Three Dimensional Cone Filter Banks", IEEE Transactions on Circuits and Systems – 1, Fundamental Theory and Applications, Vol. 50, No. 2, February 2003, pp. 208-216.
- [3] Bruton L.T and S. Singh, "Plane Wave Filtering Using A Novel 3D Cone -Stop Filter Bank", Circuits and Systems, MWSCAS-2002, 45<sup>th</sup> Midwest Symposium on, Vol. 3, August 2002, pp. 676-679.
- [4] R. K. Bertschmann, N.R. Bartley and L.T. Bruton, "A 3-D Integrator Differentiator Double Loop (IDD) Filter For Raster-Scan Video Processing", Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on, Vol. 1, 28 April-3 May 1995, pp. 470-473.
- [5] Kulach C.J., Bruton L.T. and Bartley N.R., "A Real-Time Video Implementation Of A Three-Dimensional First-Order Recursive Discrete-Time Filter", IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 699 –702.

## APPENDIX

### Algorithm 1: Hardware Mapped 1D Algorithm to Implement the First-order 2D IIR Frequency Planar Filter

```

k = 0 ; // Filter starts up here
SDPn1 = 0 ; SDPn2 = 0 ; SDPn3 = 0 ; SDPn4 = 0 ;
Address = 0 ; // All the above are global variables
do
{
Address_Analog_Multiplexer ( ) ;
wSCAN(k)=Read_Scanned_Input_Port();

w00 = wSCAN(k);
w10 = SDP_Block_Function1(wSCAN(k-1)) ;
w01 = wSCAN(k-N1);
w11 = SDP_Block_Function2(wSCAN(k-N1-1)) ;

y10 = SDP_Block_Function3(ySCAN(k-1)) ;
y01 = ySCAN(k-N1);
y11 = SDP_Block_Function4(ySCAN(k-N1-1)) ;

ySCAN(k) = a00w00 + ... a11w11 - b10y10 ... -b11y11 ;
// Calculate 2D convolution in real-time
// Using Vector dot product parallel processor

k++; // Next asynchronous clock pulse
} while (POWER_ON) ;

```

```

float SDP_Block_Function1 (float SDPinput) // Only 1 SDP algo
// is shown

```

```

{
float Output_Value = 0;
Output_Value = (SDPn1==0) ? 0: SDPinput ; // If
// Spatial ZIC is required
// connect zero to output
// port. Else connect SDP
// input to SDP output

SDPn1++; // Increment internal counters
If (SDPn1==N1) SDPn1 = 0 ; // Reset counter
// when n1=N1

Return Output_Value ;
}

```

### Algorithm 2: Hardware Mapped Algorithm For Selecting An Analog Input Channel From An Array Of Sensors Depending On The Location Of The Spatial Recursion

```

Address_Analog_Multiplexer ()
{
Write_To_Hardware_Port ( Address ) ;
Address++;

If (Address==N1) Address=0 ;
}

```

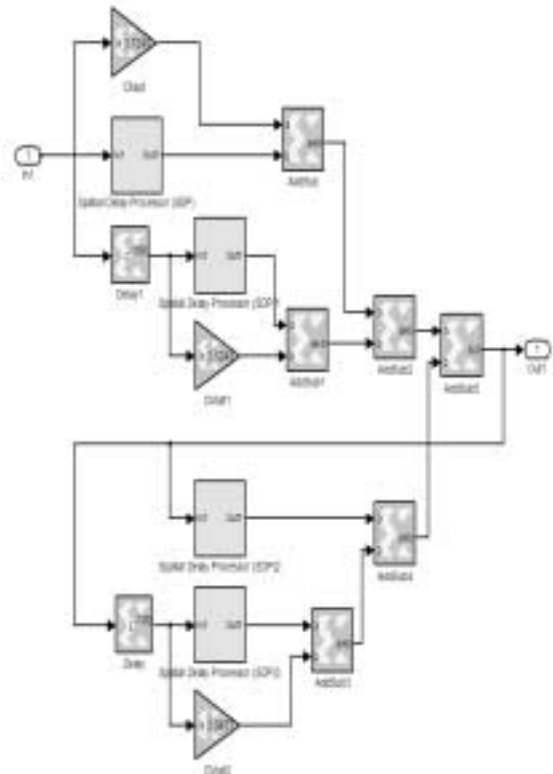


Fig. 5 Xilinx Filter Mask Implemented With XSG.