IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

CORDIC based Architecture of FFT for Orthogonal Frequency-Division Multiplexing Receiver

Sushma. U. Bhover[1], Mehrunnisa .S.P[2]

Post-Graduate Scholar, Department of EIE, DSCE, Bengaluru, India[1]

Assistant Professor, Department of EIE, DSCE, Bengaluru, India[2]

Abstract: This project is an attempt to develop a MIMO based Orthogonal Frequency Division Multiplexing (OFDM) using a CORDIC based FFT implementation. FFT acts as the very important block of the OFDM communication. As the computations are carried out on the hardware for any communication algorithms the computational complexity has to be low. A Multi Input Multi output OFDM implementation is carried out using the CORDIC based FFT algorithm in order to reduce the computational complexity in the OFDM algorithm. Matlab based implementation is carried out and the results are tabulated and discussed. The analysis of this algorithm is carried out with performance analysis.

Keywords: CORDIC, FFT, MIMO, OFDM.

I. INTRODUCTION

In the last few years wireless communications have experienced a fast growth due to the high mobility that they allow. However, wireless channels have some disadvantages like multipath fading that make them difficult to deal with. A modulation that efficiently deals with selective fading channels is OFDM. There are a large number of FFT algorithms and architectures in the signal processing literature. Therefore, the state of art algorithms and architectures should be analysed and compared. Based on different algorithms and architectures, different power consumptions, area and speed of the processor will be achieved. So their ASIC suitability should be analysed and the effort should be focused on the choosing algorithms and architectures and optimization.

The WLAN is used to provide an extremely high performance radio-access technology that offers high spectral efficiency and also provide high throughput data. Because of scalable bandwidth, operators will be able to easily migrate their networks. The WLAN receiver systems with multiple transmit and receive antennas (MIMO) present a better performance on two different angles, the diversity and the multiplexing. MIMO exploits higher transmission rates, higher spectral efficiencies, greater coverage, improved link robustness, without increasing total transmission power or bandwidth [1]. But, the WLAN receiver also increases the computational and the hardware complexities greatly.

It is a challenge to realize the receiver with the MIMO OFDM system with minimal hardware complexity and power consumption—especially the computational complexity. The FFT is the highest computational complexity modules in the baseband of WLAN Standards. Radix–r FFT algorithms are efficient realization techniques of FFT operations. So, here radix-2 based on Decimation in Time Algorithm [2] is being used. In the conventional FFT butterfly structure, each butterfly unit is generally realized by complex multipliers and adders, together with the stored required twiddle factors in memory incur high area overhead.

In order to reduce these overheads, a CORDIC algorithm and its architectures specifically tailored for FFT computations which will consume smaller areas than conventional multiplier-based FFT units. Also, a reconfigurable based FFT structure has been realized to improve the performance while retaining the software flexibility. Here static reconfiguration is been used. Since the value of N which takes the value of 2/4/8/16/32/64 during the compile time to produce the corresponding FFT outputs. This CORDIC based FFT is also used to realize the physical layer of MIMO-OFDM based IEEE 802.11n standard with minimal computation and hardware complexity. This approach has the advantages of design simplicity and high-speed operations.

II. LITERATURE SURVEY

Bevan M Bass (1999)[1] presented an energy-efficient, single-chip, 1024-point fast Fourier transform (FFT) processor. The FFT processor used cached memory architecture which offers increased speed and increased energy efficiency. Eight parallel basic processing modules in the entire chip which can work at the same time independently which can

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

compute 4096 complex point 22 forward and inverse FFT in real time with high throughput was designed by Yongjun Peng (2003).

Yun-Nan Chang et al (2003)[5] presented an efficient VLSI architecture of the pipeline fast Fourier Transform processor based on radix-4 decimation-in-time algorithm with the use of digit-serial arithmetic units. The architecture which combined both feed-forward and feedback commutator schemes achieved 100% hardware utilization and required much less memory. The overall ROM size was reduced by a factor of 2 by exploiting the redundancy of the twiddle factors.

A low power consumption and small area FFT processor architecture suitable for OFDM demodulators was introduced by Xiaojin Li et al (2007)[6]. In order to meet the requirements of high-speed data throughput, low power and small area consumption, distributed memory architecture was developed. This FFT architecture could meet the requirement of OFDM demodulators in DVB-T and other high speed wireless applications. Jesus Garcia et al (2007) have used an architecture for FPGA implementation of a Split-Radix FFT processor, which combines the higher parallelism of the 4r-FFTs and the possibility of processing sequences having length of any power of two. The simultaneous operation of the multipliers and adder-subtractors implicit in Split-radix FFT lead to faster operation.

Chao Cheng et al (2011)[3] utilized the Multi-path delay commutator structures to improve the throughput rate of radix-2 and radix-4 FFT computation by a factor of 2 to 4. Latency was reduced by a factor of 2 to 3 by saving delay elements at the cost of the number of multiplier and adders. The high-throughput FFT doubled the throughput rate when compared to previous radix-2 and radix-4 FFT structures. Although split radix FFT design is more hardware efficient, the regular structure of the FFT structures is attractive for high throughput FFT design.

III. CORDIC ALGORITHM

CORDIC (COordinate Rotation DIgital Computer) also known as Volder's algorithm, is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions, typically converging with one digit (or bit) per iteration. It is therefore also a prominent example of digit-by-digit algorithms. CORDIC and closely related methods known as pseudo-multiplication and pseudo-division or factor combining are commonly used when no hardware multiplier is available (e.g. in simple microcontrollers and FPGAs), as the only operations it requires are addition, subtraction, bitshift and table lookup. As such, they belong to the class of shift-and-add algorithms.

CORDIC can be considered fast by two facts: (1) When we take a binary number and multiply it by 2^n, we shift the binary point n places to the right and when we divide by 2^n we shift the binary point n places to the left. (2) The operations on a computer that are cheapest and fastest to perform are (A) addition and subtraction, (B) comparing numbers to see which is larger or smaller, (C) storing and retrieving numbers from memory, and (D) shifting the binary point. Addition and subtraction are very fast, but not multiplication or division. The machine does compute multiplications and divisions by powers of 2 very quickly however, by just shifting the binary point. CORDIC exploits this, and thus uses only operations (A)–(D) to evaluate sines and cosines.

1. Rotation method

To rotate a point P = (x_0, y_0) through an angle of θ degrees. The rotated point has coordinates P1 = (x, y) where

$$x = x_0 \cos\theta - y_0 \sin\theta,$$

$$y = x_0 \sin\theta + y_0 \cos\theta,$$

which can be written in matrix form as $P_1 = R_\theta P_0$ where

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

Now suppose we rotate the point (1, 0) by θ radians. Then the rotated point, P_1 = (x, y) has coordinates (cos θ ,sin θ) CORDIC breaks down rotation by θ radians, into rotations by smaller angles θ_0, θ_1, θ_2, θ_3,.... These are cleverly chosen so that these rotations can be calculated using angles hardwired into the computer. It then generates points P_1, P_2, P_3 etc. on the unit circle which approach the point (cos θ ,sin θ).

The angles whose tangents are of the form $1\ 2^n$, where n = 0, 1, 2, 3,.... The list of θi's:

$\theta_0 = \tan{-}1\ (1) = \pi/4 \approx .785398,$
$\theta_1 = \tan{-}1\ (1/2) \approx .463648 \approx 1/2,$
$\theta_2 = \tan{-}1\ (1/4) \approx .244979 \approx 1/4,$
$\theta_3 = \tan{-}1\ (1/8) \approx .124355 \approx 1/8,$
$\theta_4 = \tan{-}1\ (1/16) \approx .062419 \approx 1/16.$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

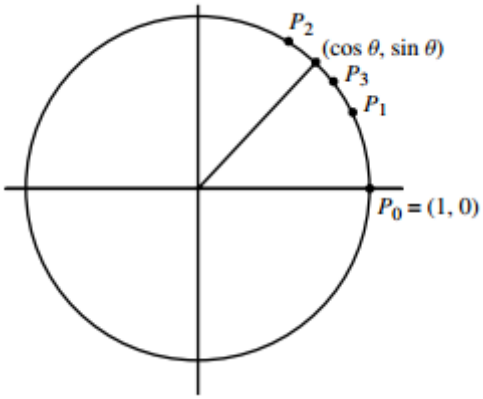

Fig.1 Unit circle

We observe that when θ is small, tan θ ≈ θ. Thus, we don't even need to store all the values θi , since eventually tan θ and θ are the same for practical purposes.

The angle θ = 1 radian, and try to build it from the θi's we have chosen. We start with $z_0 = \theta_0 = .785398$. This is too small, so the algorithm now adds θ_1 attempting to bring the total up to 1 radian. We get

$$z_1 = \theta_0 + \theta_1 \approx .785398 + .463648 = 1.249046$$

This is more than 1 radian, so the algorithm backs up. It subtracts θ_2, to get

$$z_2 = \theta_0 + \theta_1 - \theta_2 \approx 1.249046 - .244979 = 1.004067.$$

This is still more than 1, so the algorithm backs up again

$$z_3 = \theta_0 + \theta_1 - \theta_2 - \theta_3 \approx 1.004067 - .124355 = 0.879712.$$

This is less than 1, so we add

$$\theta_4 = .062419,$$

$z_4 = \theta_0 + \theta_1 - \theta_2 - \theta_3 + \theta_4 \approx .879773 + .062419 = 0.942192$, and so on.

With 40 terms, which is what many machines use,

$$z_{39} = \theta_0 + \theta_1 - \theta_2 - \theta_3 + \theta_4 + \theta_5 + \theta_6 + \theta_7 + \theta_8 + \cdots - \theta_{39}. \qquad (2)$$

At each step, the machine checks to see if the running total z_i is more or less than the angle of 1 radian, and either adds or subtracts the next θ accordingly. Simultaneously, the algorithm performs the corresponding rotations. So, following (2) we initially rotate (1, 0) by θ_0, then by θ_1, then rotate by $-\theta_2$, then rotate by $-\theta_3$ and so on. After 40 rotations, we get

$$P_{40} = R_{-\theta 39} \cdots\cdots R_{-\theta 2} \cdot R_{\theta 1} \cdot R_{\theta 0} \cdot P_0$$

$$= \begin{pmatrix} \cos(-\theta_{39}) & -\sin(-\theta_{39}) \\ \sin(-\theta_{39}) & \cos(-\theta_{39}) \end{pmatrix} \cdots \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \qquad (3)$$

Each matrix in the above product (3) is of the form

$$R_{\theta_i} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}$$

which can be rewritten as

$$R_{\theta_i} = \cos\theta_i \begin{pmatrix} 1 & -\tan\theta_i \\ \tan\theta_i & 1 \end{pmatrix}, \qquad (4)$$

and now we see how the tangent function comes into play. Since by design, $\tan\theta_i = 1/2_i$, (4) becomes

$$R_{\theta_i} = \cos\theta_i \begin{pmatrix} 1 & -\frac{1}{2^i} \\ \frac{1}{2^i} & 1 \end{pmatrix}. \qquad (5)$$

(Note: R−θi, is of the same form with off diagonal entries switched.) Finally, since for $-\pi/2 \le \theta \le \pi/2$, $\cos\theta = \sqrt{1/1+\tan 2\,\theta} = \sqrt{1/1+(2^{-i})^2}$, (5) can be written as

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

$$R_{\theta_i} = \frac{1}{\sqrt{1+(2^{-i})^2}} \begin{pmatrix} 1 & -\frac{1}{2^i} \\ \frac{1}{2^i} & 1 \end{pmatrix}$$

Now substituting into (3), we get

$$P_{40} = \frac{1}{\sqrt{1+(2^{-39})^2}} \begin{pmatrix} 1 & \frac{1}{2^{39}} \\ -\frac{1}{2^{39}} & 1 \end{pmatrix} \cdots \frac{1}{\sqrt{1+(2^{-1})^2}} \begin{pmatrix} 1 & -\frac{1}{2^1} \\ \frac{1}{2^1} & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{1+(2^0)^2}} \begin{pmatrix} 1 & -\frac{1}{2^0} \\ \frac{1}{2^0} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Pulling the constants, $\sqrt{1}\frac{1}{1+(2-i)2}$ to the front of this matrix product, we get that

$$P_{40} = K \begin{pmatrix} 1 & \frac{1}{2^{39}} \\ -\frac{1}{2^{39}} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & -\frac{1}{2^1} \\ \frac{1}{2^1} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2^0} \\ \frac{1}{2^0} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Where,

$$K = \frac{1}{\sqrt{1+(2^{-39})^2}} \frac{1}{\sqrt{1+(2^{-38})^2}} \cdots \frac{1}{\sqrt{1+(2^{-1})^2}} \frac{1}{\sqrt{1+(2^0)^2}},$$

and we are on our way. K is a constant! Once we compute it, we can, and do, hard-wire it into the machine. It turns out that K ≈ .607252935. So finally (3) becomes

$$P_{40} \approx .607252935 \begin{pmatrix} 1 & \frac{1}{2^{39}} \\ -\frac{1}{2^{39}} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & -\frac{1}{2^1} \\ \frac{1}{2^1} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2^0} \\ \frac{1}{2^0} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad (6)$$

Observe that all entries in (6) are hard-wired into the machine. No computation is needed to obtain them. Then, as we shall see, multiplying the matrices is very fast.

IV. CORDIC BASED FFT ARCHITECTURE FOR OFDM

The overall structure of the proposed CORDIC-based FFT processor model is shown in Figure 2. The entire model is made of the address generation unit, the control unit, the dual port RAM unit, the 4-point butterfly unit and the CORDIC twiddle factor generation unit. This model is characterized by setting the parameter, sampling points and the accuracy to meet the actual needs. The FFT processor presented here is based on radix-4 DIT algorithm in which the in-place computation is utilized to achieve an efficient use of the memories. To perform these operations concurrently, a dual-port RAM has been employed.

The control unit involves the timing control of the data storage, reading and writing to make the corresponding data and rotating factor coefficient flow into the butterfly and CORDIC computing unit in sequence in FFT operation. Data and addresses of the 'twiddle factor' can be easily generated by the counter. The address generation logic is very simple and does not limit the throughput of the system

The CORDIC processor performs the vector rotation to compute a set of trigonometric functions. The principle idea of the CORDIC algorithm is to decompose a rotation into a sequence of micro-rotations. The pipelined CORDIC arithmetic unit can be obtained by decomposing the CORDIC algorithm into a sequence of operational stages. The corresponding stages are expressed by the following iterative.

The CORDIC method can be employed in two different modes, known as the rotation mode and the vector mode. In this paper, one uses x and y to represent the real part and imaginary part of input and output vectors. In the rotation mode, the co-ordinate components of a vector and an angle of rotation are given and the co-ordinate components of the original vector are computed after rotation by a given angle.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

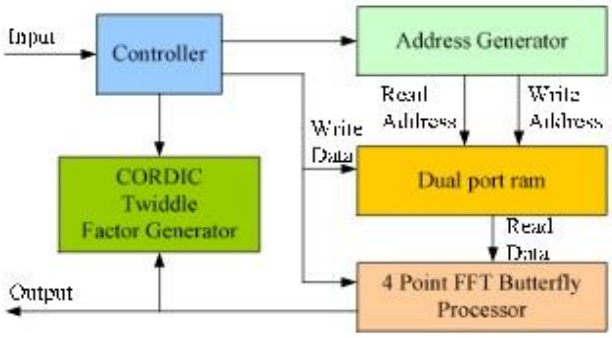

Figure 2. Proposed CORDIC -based FFT architecture

In the vector mode, the co-ordinate components of a vector are given and the magnitude and angular argument of the original vector are computed.
All the iterations of CORDIC algorithm are performed in parallel by using a pipelined structure [9], as shown in figure 3. The pipelined structure ensures the highest possible throughput, because a CORDIC transformation can be performed in each clock cycle.

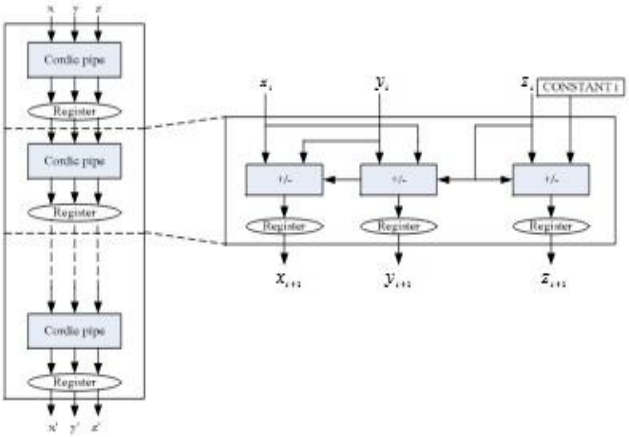

Figure 3. CORDIC unit architecture

V. SIMULATION RESULT

The proposed design for the CORDIC-based radix-4 FFT processor has been realized by HDL and implemented with an FPGA chip (Virtex 6) on Quartus-II 9.1 platform. The requirements of harmonic analyzer, the design and implementation of high-speed FFT processor is the most important work. Due to finite precision and the accuracy to meet the actual needs, the sampling points is 256,which are processed in 5 stages by the FFT processor. The word length of the complex fixed-point data computed by butterfly processor is 16-bit. And the CORDIC operator is implemented pipelined structure of 16 stages.

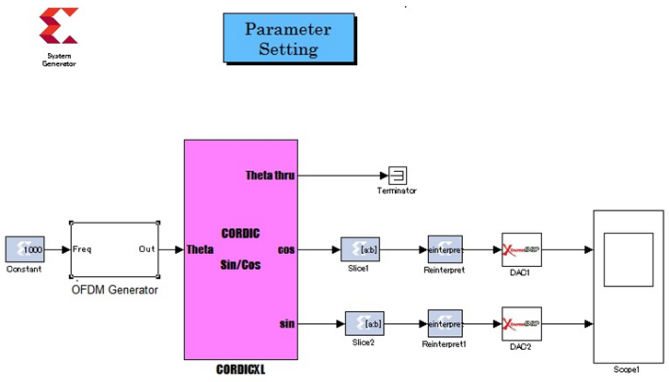

Figure 3 Simulink Block diagram of CORDIC based FFT for OFDM

DOI10.17148/IJARCCE.2017.6652

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

Figure 3 is the Simulink block diagram of the CORDIC based FFT for OFDM receiver and the area and the power consumption of the Virtex 6. The Simulink file is linked with the Xilinx to get the output of the area reduction in the system generator kits used.

The Synthesis report can be simulated to find out the number of Adders shifters used, LUT used by the FPGA kit Virtex 6

```
Number of Slices:                494  out of  33280    1%
Number of Slice Flip Flops:      420  out of  66560    0%
Number of 4 input LUTs:          773  out of  66560    1%
   Number used as logic:         767
   Number used as Shift registers:  6
Number of IOs:                    44
Number of bonded IOBs:            43  out of    633    6%
Number of GCLKs:                   1  out of      8   12%
```

Device	
Family	Virtex6
Part	xc6vlx240t
Package	ff1759
Temp Grade	Commercial
Process	Typical
Speed Grade	-2

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.006	1	--	--
Logic	0.003	575	150720	0
Signals	0.002	803	--	--
IOs	0.001	43	720	6
Leakage	3.689			
Total	3.702			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	1.682	0.011	1.671
Vccaux	2.500	0.135	0.000	0.135
Vcco25	2.500	0.002	0.000	0.002
MGTAVcc	1.000	0.910	0.000	0.910
MGTAVtt	1.200	0.638	0.000	0.638

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA
Board Temperature (C)	NA

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	1.1	80.8	54.2

Supply Power (W)	Total	Dynamic	Quiescent
	3.702	0.013	3.689

VI. CONCLUSION

In this paper, the design of a CORDIC algorithm based radix-4 FFT processor for OFDM using FPGA with its intended application in power signal processing is presented. The choice of the CORDIC algorithm for realizing the basic butterfly operation for the FFT which does to reduces the hardware complexity. This kind of architecture can solve the contradiction between the real time and accuracy in the detection of the power.

REFERENCES

[1] Jos. Arrillaga, Neville. R. Watson, Power System Harmonics, Second Edition, CHINA ELECTRIC POWER PRESS, 2008.
[2] LIU. Min,WANG. Ke. ying "A high accurate harmonic analysis method based on FFT and neural network in power system", Electric Power College,South China University of Technology, Guangzhou, 510640,China
[3] SHA. Zhan.you, New dedicated digital instrumentation and application of principles, Beijing, China Machine Press, 2006, pp. 89-93.
[4] ZHU Xiao.hong, The electric energy measurement, China Electric Power Press, 2007, pp.38-57.
[5] ZHANG. Qing. heng, "The Multi-Function Electric Power Meter Based on the DSP",Hangzhou,Zhe Jiang University,2007, pp.48-53.
[6] Li. Cheng. shi, Chu. Jian. peng, Li. Xin. bing, "A High Speed Real-time and Fixed-point FPGA Realization of a CORDIC Based FFTProcessor", Microelectronics & Computer, Apirl 21th, 2004, pp.88-91.
[7] J. Volder, "The CORDIC trigonometric computing technique", IEEE Transactions on Electronic Computers, vol. EC-8, no. 8, pp. 330-334, September 1959.
[8] Hong. Zhi. Wang, Louet. Y, Palicot. J, Alaus. L.; Noguet. D, "Memory-efficient FFT architecture using R-LFSR based CORDIC common operator", Cognitive Information Processing (CIP), 2010 2nd International Workshop. 2010 , pp.162 - 167
[9] Richard. Herveille,"Cordic Core Specification". unpublished.
[10] Zhao. Ming,"radix 4 complex fft". unpublished