Pipeline direct digital frequency synthesiser using decomposition method

S.-I.Liu, T.-B.Yu and H.-W.Tsao

Abstract: A direct digital frequency synthesiser using a new decomposition method without the large sine ROM table is presented. To improve its operating frequency a pipeline structure has been utilised. It has been fabricated in a 0.6μm single-poly double-metal (SPDM) CMOS process and its core area is $0.95 \times 1.1\,mm^2$. The maximum operating frequency is 85MHz. For a 10MHz sinusoidal output, the phase noise is –114dBc/Hz at an offset frequency of 10kHz. The measured SNR is 60.77dB and worst case spurious is –67.6dBc. Its power dissipation is 80mW at 80MHz under the 5V supply.

1 Introduction

A frequency synthesiser is one of the important building blocks for wireless communication. There are two major methods of coherent frequency synthesis, direct and indirect [1]. The indirect methods are primarily phase-locked loop (PLL)-based. Direct digital frequency synthesiser (DDFS) methods can have many important advantages over the PLL-based methods, including smaller frequency resolution, fast and phase-continuous frequency switching and excellent temperature and ageing stability etc. However, DDFS methods may require larger chip area and power consumption. Most DDFSs use the sine table look-up method [2–4]. This method requires a large ROM to store the sinusoidal samples, although many improved methods [3–7] utilise the symmetrical property and trigonometric decomposition of the sine function to reduce the ROM size. In addition, some other methods such as the Taylor series approximation [8], CORDIC algorithm [9] and residue number [10] computation methods have also been presented to implement the direct digital frequency synthesiser. The Taylor series approximation method needs three small-size ROMs and large computing complexity for multiplication and the power dissipation will be large for high frequency sine wave synthesis. The CORDIC method can reduce the required ROM size efficiently, but the computation complexity is also large. The residue number method does not need any multiplication, but many small ROMS and a residue number system to binary conversion circuit are required. In this paper, new decomposition and pipeline methods are used to create a DDFS without large sine ROM tables. It has the following advantages: no multiplication, no large sine ROM, low computation complexity and low power dissipation.

IEE Proceedings online no. 20010158

DOI: 10.1049/ip-cds:20010158

Paper first received 8th October 1999 and in revised form 6th November 2000

The authors are with the Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan 10617, Republic of China

2 Circuit description

A conventional DDFS consists of a phase accumulator, a sine ROM look-up table and a digital-to-analogue converter (DAC). A frequency control word is applied to the phase accumulator to produce the address word for the sine ROM table. The output of the ROM table is converted to a sinusoidal signal by a DAC. The proposed DDFS uses a new decomposition method. A sine function is decomposed into the summation of two functions: coarse function $f(x)$ and fine function $g(x)$ as shown in Fig. 1. Such a sinusoidal decomposition method can be expressed as

$$\sin x = f(x) + g(x) \tag{1}$$

where x is the output of the phase accumulator. According to the required signal-to-noise ratio (> 60dB), a 14-bit address word at the output of the phase accumulator and 10-bit outputs for the DAC are needed. For the coarse function as shown in Fig. 1, a quarter-wave sine waveform with the amplitude of 4095 is divided into eight sections with the magnitude of every section being the exponent of 2 (i.e. $2^9 = 512$). If the sine function is expressed in binary form, the first three most significant bits just represent the coarse function $f(x)$ and the remaining bits are the fine function $g(x)$. One of advantages for this kind of decomposition is that the coarse and fine functions are independent and the output of sin x can be obtained without extra

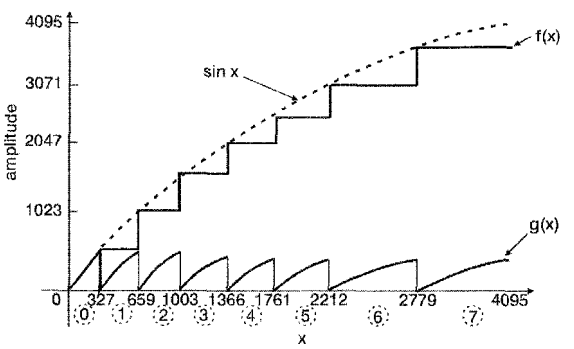

Fig. 1 *Decomposition of sine function*

adders to sum $f(x)$ and $g(x)$. Moreover, according to Table 1, depending on the value of the output bits of the phase accumulator, $f(x)$ can be implemented using only some combinational logic gates. For example, if x falls within the interval of [0, 326] (the 0th section in Table 1), the three most significant bits will be 000. In fact, only 75 gate counts are needed to realise $f(x)$ in this DDS.

Table 1: Coarse function *f(x)*

Section	Address: x	Operation	$f(x)$: three MSB bits of sin$x$
0	[0:326]	$0 <= x < 327$	000
1	[327:658]	$327 <= x < 659$	001
2	[659:1002]	$659 <= x < 1003$	010
3	[1003:1365]	$1003 <= x < 1366$	011
4	[1366:1760]	$1366 <= x < 1761$	100
5	[1761:2211]	$1761 <= x < 2212$	101
6	[2212:2778]	$2212 <= x < 2779$	110
7	[2779:4095]	$2779 <= x < 4095$	111

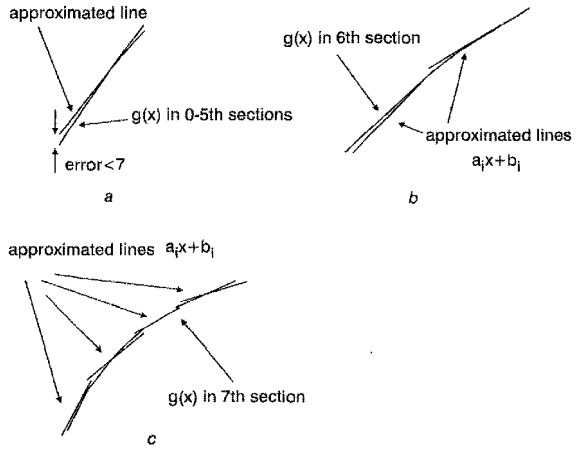

Fig.2 *Approximation of fine function g(x) using line segments*

Table 2: Fine function, $g(x) = a_i \cdot x - b_i$ where a_i and b_i are constants

Section i	Address: x	$a_i \cdot x - b_i$ (decimal)	$a_i \cdot x - b_i$ (binary)
0	[0:326]	$1.5625 \cdot x$	$1.1001_2 \cdot x$
1	[327:658]	$1.53125 \cdot x - 497$	$1.10001_2 \cdot x - 497_{10}$
2	[659:1002]	$1.4375 \cdot x - 939$	$1.0111_2 \cdot x - 939_{10}$
3	[1003:1365]	$1.40625 \cdot x - 1409$	$1.01101_2 \cdot x - 1409_{10}$
4	[1366:1760]	$1.28125 \cdot x - 1745$	$1.01001_2 \cdot x - 1745_{10}$
5	[1761:2211]	$1.125 \cdot x - 1974$	$1.001_2 \cdot x - 1974_{10}$
6	[2212:2495]	$0.9375 \cdot x - 2067$	$0.1111_2 \cdot x - 2067_{10}$
	[2496:2778]	$0.8125 \cdot x - 1747$	$0.1101_2 \cdot x - 1747_{10}$
7	[2779:3071]	$0.6875 \cdot x - 1908$	$0.1011_2 \cdot x - 1908_{10}$
	[3072:3455]	$0.5 \cdot x - 1332$	$0.1_2 \cdot x - 1332_{10}$
	[3456:3839]	$0.269531 \cdot x - 536$	$0.01000101_2 \cdot x - 536_{10}$
	[3840:4095]	$0.078125 \cdot x + 194$	$0.000101_2 \cdot x + 194_{10}$

The approximation of the fine function $g(x)$ can be described as follows: form the 0th section to the fifth section of the fine function, single straight-line segments with different slope factors $a_i (i = 0 - 5)$ can be used to approximate $g(x)$ as shown in Fig. 2*a* and the error can be limited within $0 \sim 7$. In this way, $g(x)$ can be implemented using only adders and multiplexers without multipliers. However, if a single straight line is used to approach the ideal curve in the sixth and seventh sections, the error will be larger. Therefore, two line segments and four line segments are employed in the sixth and seventh sections as shown in Figs. 2*b* and *c*, respectively, to keep the error below seven. The fine function, $g(x) = a_i \cdot x - b_i$ for every section i, has been calculated and optimised by Matlab [11] and they are listed in Table 2. Therefore the sin function sin x, can be rewritten as

$$\sin x = f(x) + g(x) \cong [c_i] + [a_i x - b_i] \qquad (2)$$

where a_i and b_i are the coefficients listed in Table 2 and c_i is the output of the coarse function listed in Table 1 and i is the section number. To see how operation of eqn. 2, one may rewrite it as

$$\sin x = f(x) + g(x) \cong [C_i] + [A_i x + D_i] \qquad (3)$$

where $A_i = a_i$, $C_i = c_i$ and $D_i = 1$ + the complement of b_i. Investigating the slope factors a_i in Table 2, one can find that the number of '1's for the slope factor a_i is no more than four. Thus the operation of $a_i x$ can be implemented with three adders and four shift operations but no multipliers, as shown in Fig. 3. For example, in Section 2, the fine function can be approximated as

$$\begin{aligned} g(x) &= (1.4375 \bullet x - 939)_{10} = (1.0111 \bullet x)_2 + 3156_{10} \\ &= \left(\left(x + \frac{x}{4}\right) + \left(\frac{x}{8} + \frac{x}{16}\right)\right)_2 + 3156_{10} \end{aligned} \qquad (4)$$

where the subscripts 10 and 2 denote that the numbers are represented by decimal and binary, respectively. According to eqn. 4, the multiplication of a_i and x can be realised by applying four shift operations to x and three adders. The constant can also be realised by combinational logic gates. Since the number of the slope coefficients a_i expressed in terms of binary bits is no more than four, the fine function $g(x)$ can be realised by four adders and four shift operations. The sine function sin$x = f(x) + g(x)$, can then be obtained without no extra adders.

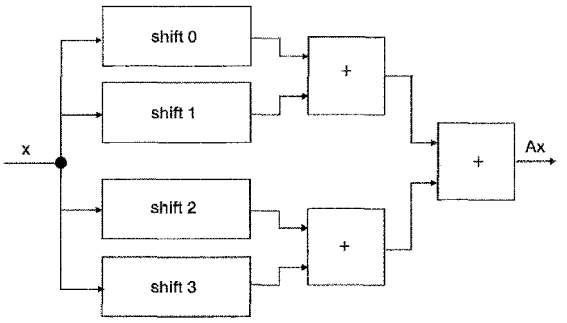

Fig.3 *Realisation of operation $a_i x$*

Based on this decomposition method, the proposed DDFS has been designed as shown in Fig. 4. To further improve the speed of this DDFS, a pipeline structure is adopted. The whole structure is divided into four parts. The first part includes the 16-bit phase accumulator and some logic circuits to initiate the orthogonal operation. In this DDFS, logic circuits are used to detect the moment at which the sine signal goes through the positive 90°. Once it goes through 90°, a flag signal is pulled high as shown in Fig. 5. This flag signal can be used to initiate the other DDFS to provide the quadrature output. The second part realises the coarse function according to Table 1. The third

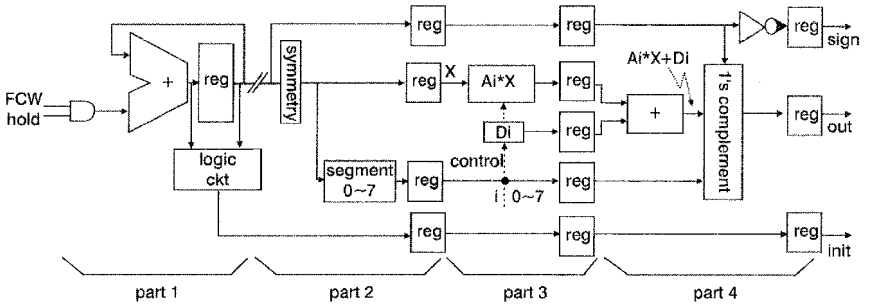

Fig. 4 *Block diagram of proposed DDFS*

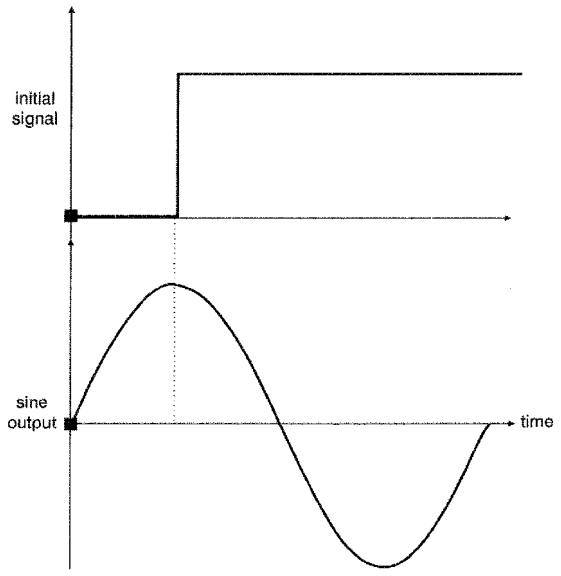

Fig. 5 *Timing diagram for realising orthogonal signals*

and fourth parts realise the fine function and output the digital sinusoidal signal using the 1's complement adder. The 16-bit adder employed in the phase accumulator is the so-called carry select adder [12, 13].

To estimate the delay time in each part of the proposed DDFS, the Synopsys tool [14] is utilised. For example, the calculated delay time for the 16-bit carry select adder in the phase accumulator is about 5.3 ns. The remaining components have also been calculated as listed in Table 3. After the post simulations, the estimated maximum operating frequency can be over 100 MHz.

Table 3: Delay time estimated by Synopsys

	1st part	2nd part	3rd part	4th part
Equivalent gate count for combinational logics	255.25	196	1076	275
Equivalent gate count for registers	132	140.25	232.5	100.75
Equivalent gate count for interconnection	0.77	0.15	1.81	0.46
Total count	388.02	336.4	1310.3	376.2
Delay (ns)	5.3	5.98	6.22	7.06

3 Experimental results

The proposed circuit has been realised by using 0.6μm standard cells [15] and fabricated in a 0.6μm SPDM CMOS process. The photograph of this chip is shown in Fig. 6. A commercial DAC (AD9721) [16] is used to obtain the analogue output waveform. The measured performance of the DDFS is listed in Table 4. The transistor count used in this DDFS is 11721 and its core area is $0.95 \times 1.10\,mm^2$ (excluding pads). The maximum operation frequency is 85 MHz. Its power consumption against the clock frequency is shown in Fig. 7. Fig. 8 shows the measured SNR against operating clock frequency. A typical spectrum of the DAC output at 10 MHz is shown in Fig. 9 where the second harmonic distortion is −85.84 dBc and the phase noise is −114 dBc/Hz at an offset frequency of 10 kHz.

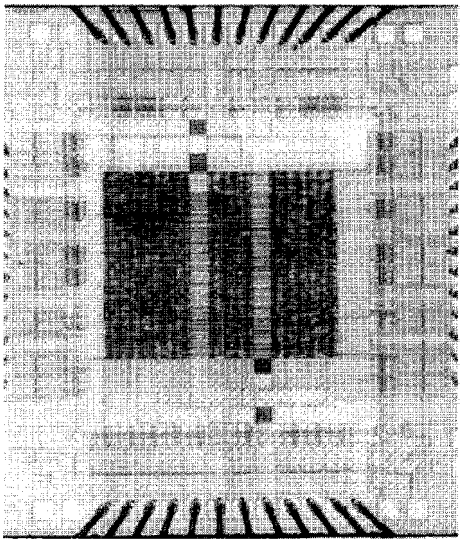

Fig. 6 *Photograph of proposed DDFS*

Table 4: Measured performances of proposed DDFS

Frequency control word	16 bits
Truncated phase control word	14 bits
Output word length	10 bits
Power supply	5 V
Power dissipation	80.7 mW @ 80 MHz
Worst-case spurious	−67.6 dBc @ 10 MHz
Phase noise	−114 dBc/Hz @ offset 10 kHz from 10 MHz
Chip Area	$0.95 \times 1.1\,mm^2$ (excluding pads)
Package	DIP 48-pin
Technology	0.6μm SPDM CMOS
Design style	cell-based
Transistor count	11721

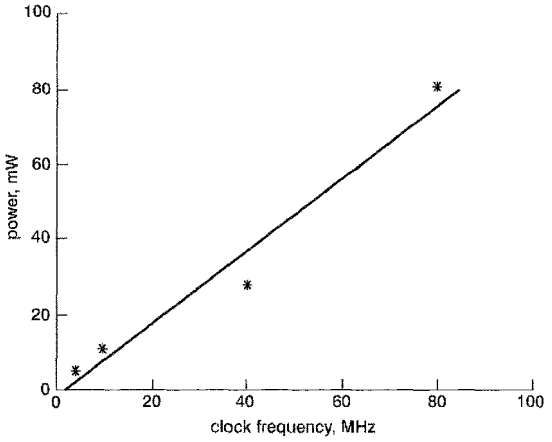

Fig. 7 *Measured power against operating clock frequency*

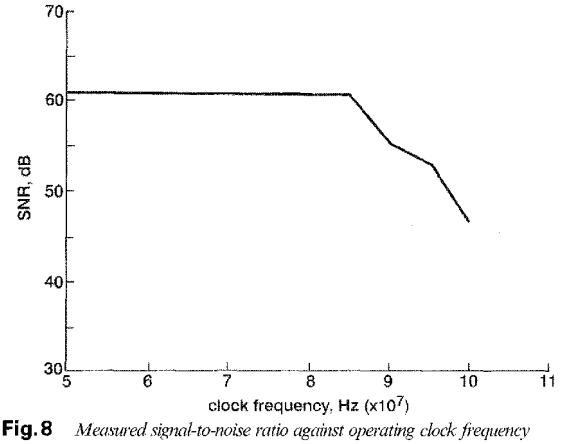

Fig. 8 *Measured signal-to-noise ratio against operating clock frequency*

4 Conclusions

A pipeline DDFS using a new sine function decomposition method has been presented and fabricated in a 0.6μm SPDM CMOS process. Using the proposed decomposition method the pipeline DDFS without the sine ROM table has been realised. The experimental results clearly demonstrate the satisfactory performance of the proposed DDFS.

5 Acknowledgment

The authors thank the National Science Council for financial support and the Chip Implementation Center (CIC), National Science Council, Taiwan, Republic of China for the fabrication of the chip. This work was sponsored by NSC-88-2215-E-002-030.

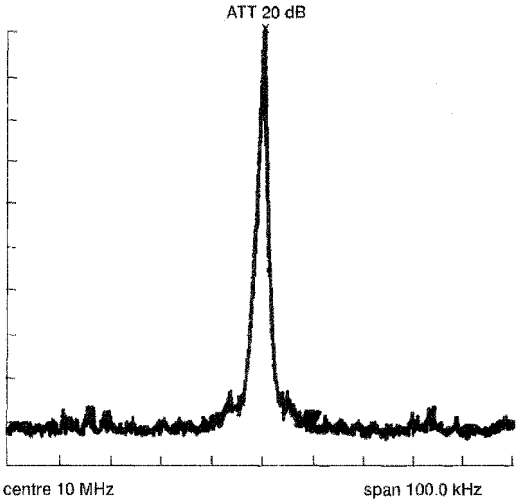

Fig. 9 *Measured spectrum of 10 MHz output of proposed DDFS*

6 References

1 MANASSEWITSCH, V.: 'Frequency synthesizers, theory and design' (Wiley, New York, 1979, 2nd edn.), p. 54
2 TIERNEY, J., RADER, C.M., and GOLD, B.: 'A digital frequency synthesizer', *IEEE Trans. Audio Electroacoust.*, March 1971, **AU-19**, pp. 48–56
3 GORSKI-POPIEL, J.: 'Frequency synthesis: techhiques and applications' (IEEE Press, New York, 1975)
4 KROUPA, V.F. (Ed.): 'Direct digital frequency synthesizers' (IEEE Press, 1999)
5 SUNDERLAND, D., STRAUCH, R., WHARFIELD, S., PETERSON, H., and COLE, C.: 'CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications', *IEEE J. Solid-State Circuits*, August 1984, **SC-19**, pp. 497–505
6 NICHOLAS, H.T., , and SAMUELI, H.: 'A 150 MHz direct digital frequency synthesizer in 1.25 μm CMOS with –90 dBc spurious performance', *IEEE J. Solid-State Circuits*, December 1991, **26**, pp. 1959–1969
7 TAN, L.K., and SAMUELI, H.: 'A 200 MHz quadrature digital synthesizer/mixer in a 0.8 μm CMOS', *IEEE J. Solid-State Circuits*, March 1995, **30**, pp. 193–200
8 Qualcomm Q2334, technical data sheet, June 1991
9 VANKKA, J.: 'Methods of mapping from phase to sine amplitude in direct digital synthesizer', *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, March 1997, **44**, pp. 526–534
10 CHREN, W.A.: 'RNS-based enhancements for direct digital frequency synthesis', *IEEE Trans. Circuits Systems. II: Analog Digit. Signal Process.*, August 1994, **42**, pp. 516–524
11 The Mathworks Inc., MA, 1996
12 UYA, M., and KANEKO, K.: 'A CMOS floating point multiplier'. IEEE ISSCC, digest of technical papers, February 1984, pp. 90–91
13 WESTE, N.H.E., and ESHRAGHIAN, K.: 'Principles of CMOS VLSI design, A systems perspective' (Addison Wesley, 1993, 2nd edn.), p. 532
14 Synopsys Inc., Mountain View, CA, 1995
15 Chip Implementation Center, National Science Council, Hsinchu, Taiwan, Republic of China
16 Analog Devices, data converter reference manual, 1992