

# Teaching Asynchronous Design in Digital Integrated Circuits

Jiann S. Yuan, *Senior Member, IEEE*, and Weidong Kuang

**Abstract**—To introduce the basis of asynchronous digital circuit design in an electrical engineering curriculum, Null Convention Logic is presented as an innovative asynchronous paradigm. The design flow from concept to circuit implementation is discussed. First, two completeness criteria are required for speed independence: symbolic completeness of expression and completeness of input. Second, threshold gates with hysteresis are primary components, which are used to build logic gates, full adder, and registers. As an example, a  $4 \times 4$  multiplier is constructed based on these threshold gates. Finally, an example of very-high-speed integration circuit hardware description language (VHDL) simulation is given to help students practice and understand the asynchronous design methodology.

**Index Terms**—Asynchronous digital circuit, Null Convention Logic (NCL), very-high-speed integration circuit hardware description language (VHDL) simulation.

## I. INTRODUCTION

DIGITAL integrated circuits have experienced rapid growth since the 1950s. This growth comes from reduced transistor dimensions, increased transistor counts, and increased operating frequencies. Further improvements for synchronous digital systems will be limited by the clock skew and excessive power consumption problems resulting from the use of a very-high-frequency system clock synchronizing the whole system. On the other hand, the significance of asynchronous circuit is becoming more and more evident as the operating speed and system complexity increase. There have been several attempts to demonstrate the potential advantages of the asynchronous circuit over its synchronous counterpart. The University of Manchester, Manchester, U.K., designed the asynchronous microprocessor [1], an embedded system chip incorporating a 32-b advanced RISC machines (ARM)-compatible asynchronous core, a cache, and several other system functions. A group at the California Institute of Technology, Pasadena, designed an asynchronous version of the million instructions per second (MIPS) R3000 [2]. Philips Research Laboratory, Eindhoven, The Netherlands, designed an 80C51 microcontroller [3] and an error corrector for a digital compact cassette player [4], which consume significantly less power than their synchronous counterparts and also have superior electromagnetic (EM) compatibility properties.

However, synchronous circuits predominate in spite of the potential advantages. One of the reasons is that asynchronous circuits are more difficult to design than synchronous circuits. In a

synchronous system, a designer can simply define the combinational logic necessary to compute the given functions and surround it with registers. In contrast, complicated design methodologies are needed to design reliable asynchronous circuits. The topic of asynchronous circuit design is usually ignored in most university curricula. One of the reasons is a lack of textbooks for asynchronous design at the undergraduate level. Although some successful asynchronous methodologies have been developed in recent years [5], this course is not typically in an undergraduate curriculum. It is, however, important to include basic concepts and simple examples on asynchronous circuit design in the initial digital logic design course to prepare for the advanced course.

This paper presents a basis of asynchronous digital circuit design to senior students and first-year graduate students. The paper is organized as follows. Section II briefly introduces the main differences between asynchronous and synchronous circuits. In Section III, the Null Convention Logic (NCL) [6] design methodology is presented to demonstrate how to design a specific asynchronous circuit in practice. NCL was patented as an innovative asynchronous paradigm in 1994 and developed by Theseus Logic, Inc., Orlando, FL. This technology is a good example for teaching purposes. After completing this course, students will gain a basic understanding of asynchronous digital circuits and achieve the learning objectives, such as the ability to define concepts of NCL and threshold gate, the ability to apply NCL to the design of standard logic functions (INVERT, AND, OR, XOR) and simple arithmetic logic unit (ALU) systems (ADDER, MULTIPLIER), the ability to define and supply the two completeness criteria of NCL, and the ability to implement NCL and hysteresis using a hardware description language (HDL). Section IV is the conclusion.

## II. DIFFERENCES BETWEEN SYNCHRONOUS AND ASYNCHRONOUS CIRCUITS

Logic design styles can be classified into two categories, namely, synchronous and asynchronous. Most modern digital systems are synchronous since they use a single rhythm. In synchronous circuits, a clock distribution system delivers the timing signals from the crystal oscillator to the various circuits. Events occur at a discrete time defined by the clock distribution system.

To describe how asynchronous systems work, Sutherland [7] used the metaphor of the bucket brigade, as shown in Fig. 1. A synchronous system is like a bucket brigade in which each person follows the tick-tock rhythm of a clock. When the clock ticks, each person pushes a bucket forward to the next person down the line. When the clock tocks, each person grasps the

Manuscript received April 29, 2003; revised September 1, 2003.

The authors are with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA.

Digital Object Identifier 10.1109/TE.2004.825923

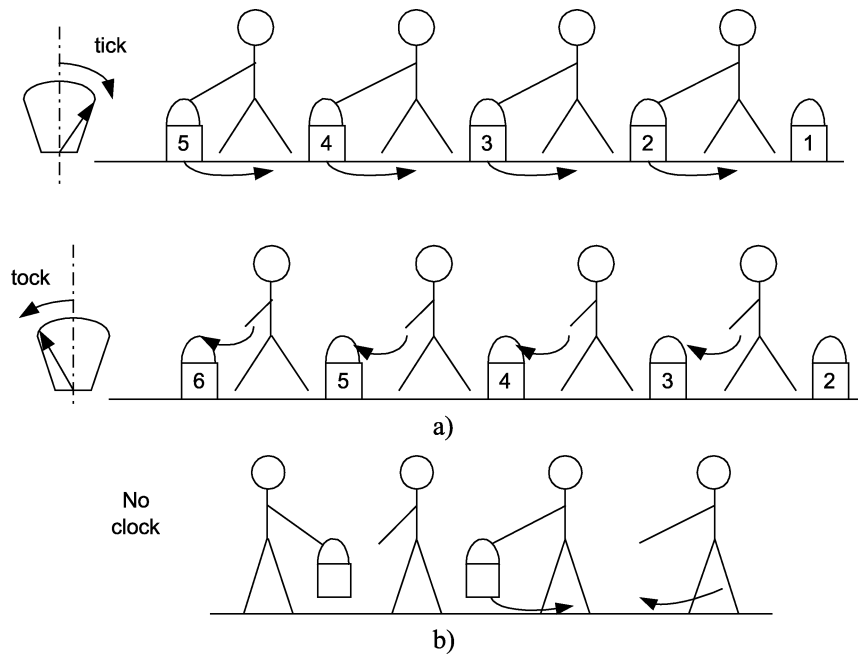


Fig. 1. Metaphor of the bucket brigade for synchronous and asynchronous circuits. (a) Synchronous. (b) Asynchronous.

bucket pushed forward by the preceding person. A synchronous system, however, does not need the clock: each person who holds a bucket can pass it down the line as soon as the next person's hands are free.

Compared with synchronous counterparts, asynchronous circuits have several potential advantages.

- 1) *Average case performance.* In a synchronous circuit, the speed of the clock is usually set according to worst-case conditions for process, temperature, voltage, and data. Although the circuit can perform quickly in some conditions, its performance is still bound by the global clock speed set for the worst case. Many asynchronous systems sense when a computation has completed, allowing them to exhibit average-case performance. For circuits where the worst-case condition is significantly worse than the average-case condition, asynchronous design can result in a substantial speed improvement. A ripple-carry adder is a good example [8].
- 2) *No clock skew.* Clock skew is the difference in arrival times of the clock signal at different parts of the circuit. Synchronous circuits often need to slow down to accommodate the skew. As feature sizes decrease and the frequency of the clock increases, clock skew becomes a much greater concern. In contrast, since asynchronous circuits have no globally distributed clock, there is no need to worry about clock skew.
- 3) *High energy efficiency.* Although asynchronous circuits often require more transitions on the computation path than synchronous counterparts, they generally have transitions only where and when involved in the current computation. Some asynchronous implementations inherently eliminate glitches, therefore decreasing energy consumption.
- 4) In mixed-signal circuits, a digital subcircuit usually generates noise and/or emits EM radiation that affects the

TABLE I  
DUAL-RAIL ENCODING

Logic value	Encoding	
	$D^0$	$D^1$
DATA 1	0	1
DATA 0	1	0
NULL	0	0
Invalid	1	1

whole circuit performance. Because of the absence of a complex clock network, asynchronous circuits should have better noise and EM properties.

### III. NCL CIRCUIT DESIGN

Asynchronous circuits can be categorized by the timing models they assume into two types: fundamental-mode circuits and speed-independent circuits. Specifically, it is assumed that delay in all circuits and wires is known, or at least bounded, in fundamental-mode circuits. This timing model is the same model used for synchronous circuits. Speed-independent circuits assume that delays in circuit elements are unbounded. The circuit is called a delay-insensitive circuit when the delays in both elements and wires are assumed to be unbounded.

#### A. Speed-Independent Circuits

In synchronous circuits, the value of a wire is assumed to be correct by a given time and can be acted upon at that time. In speed-independent circuits, there is no guarantee that a wire will reach its proper value at any specific time. Therefore, the recipient of a signal needs to inform the sender when it has received the information. This function is performed by comple-

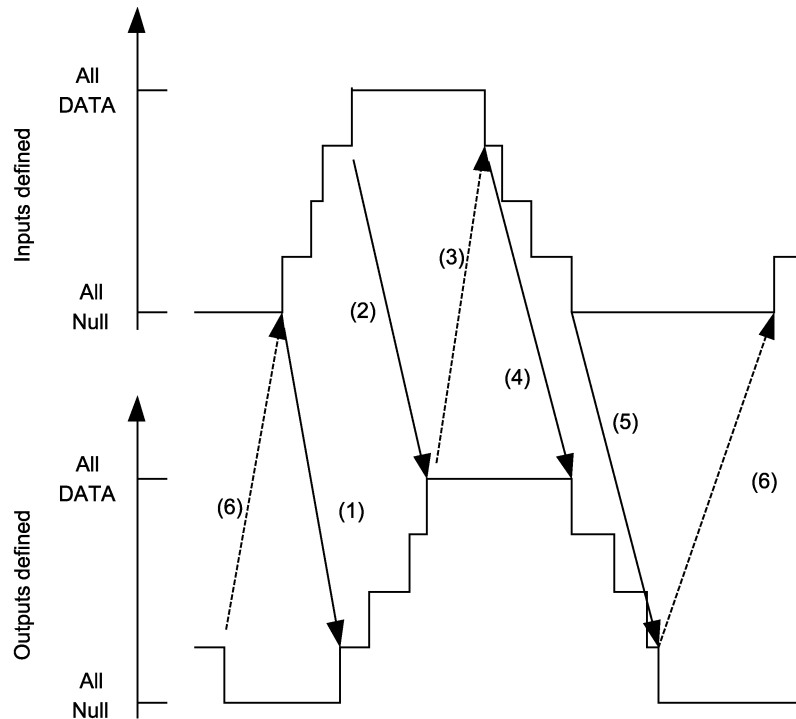


Fig. 2. Weak conditions for NCL—completeness of input.

tion detection circuitry in the receiver. The sender is required to wait until it gets the completion signal before sending the next data. A two-phase handshaking or a four-phase handshaking protocol can be used to pass data for speed-independent circuits [9]. In a two-phase handshaking protocol, a request transition is sent from the sender to the receiver to inform the validity of data and then a response transition is sent back by the completion detection logic. In a four-phase handshaking protocol, only rising transitions are used to send handshaking information, and thus the four-phase circuit structures can be simpler than their two-phase counterparts. To keep delay insensitivity (DI), multiple wires are required to transfer a data bit.

Consider an NCL implementation of the delay-insensitive circuit abstraction. Although NCL is a relatively new design paradigm, it is mature enough to be included in an electrical or computer engineering curriculum. NCL is not purely delay insensitive, but an extension of DI close to quasi-DI (QDI). NCL is based on a more general delay assumption than the isochronic fork [10], the so-called “orphan” hypotheses.

NCL uses two completeness criteria to achieve its delay-insensitive behavior: symbolic completeness of expression and completeness of input. Threshold gates with hysteresis are used to build NCL circuits. These gates are constructed directly at the transistor level, independent of basic Boolean gates. The communication protocol between modules leads to an inherent pipeline structure for NCL circuits. NCL circuits usually operate in a form of a pipeline.

*B. Completion Criteria*

The first completeness criterion is symbolic completeness of expression. A symbolically complete expression is defined as an expression that only depends on relationships of the symbols presented in the expression. Clocked Boolean logic (CBL) is not

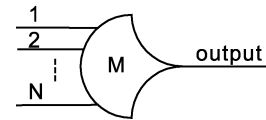


Fig. 3. Symbol of TH<sub>m</sub>n gate.

symbolically complete. In a CBL system, the binary code can use all possible code combinations, and all combinations can express valid DATA. Every wire is assumed to be independent and to assert one of two possible values, 0 or 1. All wires always assert logically valid DATA values, that is, wires cannot assert the state “not-DATA.” Therefore an external clock is needed to determine the final validity of DATA. In NCL systems, multi-rail encoding uses a subset of the possible code combinations to express valid DATA. There are mutually exclusive dependency relationships among the wires which collectively can express valid DATA, not-DATA, and illegal code. Therefore, no external clock is needed to define the validity of DATA. The control information is integrated in the data wires.

Dual-rail signals with three logic states (NULL, DATA0, and DATA1) are used to achieve symbolic completeness of expression. A dual-rail signal (D) consists of two wires D<sup>0</sup> and D<sup>1</sup>. The value of a dual-rail signal is represented by a value from the set {DATA0, DATA1, NULL}, shown in Table I. The DATA0 state (D<sup>0</sup> = 1 (or high), D<sup>1</sup> = 0 (or low)) corresponds to a Boolean logic 0. The DATA1 state (D<sup>0</sup> = 0 (or low), D<sup>1</sup> = 1 (or high)) corresponds to a Boolean logic 1. The NULL state (D<sup>0</sup> = 0 (or low), D<sup>1</sup> = 0 (or low)) corresponds to the empty set meaning that the value of D is not yet available. The state (D<sup>0</sup> = 1 (or high), D<sup>1</sup> = 1 (or high)) is forbidden.

The second criterion, completeness of input, states that for an NCL combinational circuit, 1) the output may not transition from NULL to a complete set of DATA until the input

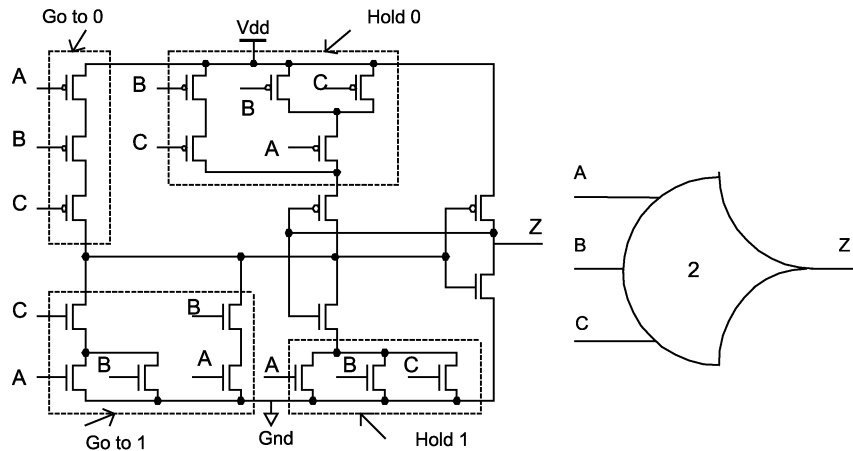


Fig. 4. Schematic and symbol of TH23.

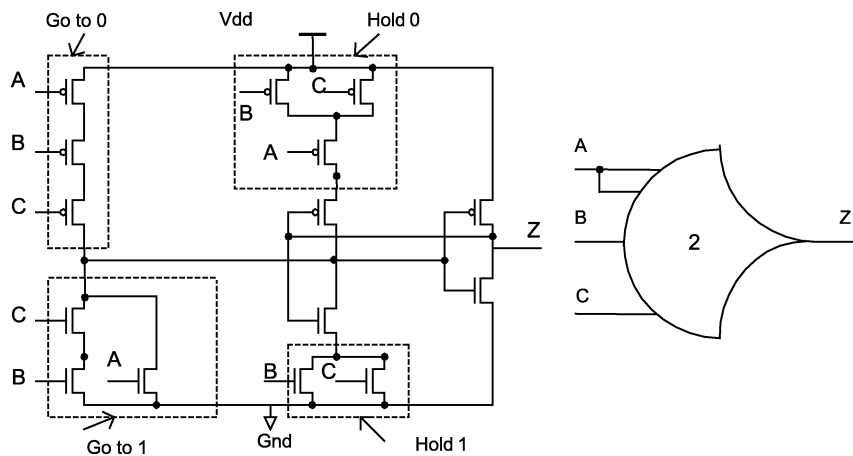


Fig. 5. Schematic and symbol of TH23W2.

values are completely DATA and 2) the output may not transition from DATA to a complete set of NULL values until the input values are completely NULL. The criterion, equivalent to Seitz's "weak condition" [8], is illustrated in Fig. 2. This criterion is a necessary condition for speed-independence.

The orderings labeled in Fig. 2 are explained hereafter.

- (1) Some inputs become DATA before some outputs become DATA.
- (2) All inputs become DATA before all outputs become DATA.
- (3) All outputs become DATA before some inputs become NULL.
- (4) Some inputs become NULL before some outputs become NULL.
- (5) All inputs become NULL before all outputs become NULL.
- (6) All outputs become NULL before some inputs become DATA.

### C. Threshold Gates With Hysteresis

NCL uses a special type of gate, namely threshold gates with hysteresis, as basic units to build NCL circuits so that NCL circuits observe the two completeness criteria.

A C-element is very desirable for the design of asynchronous circuits. Its output becomes 1 only after all of its inputs are 1,

and its output becomes 0 only after all of its inputs are 0. The threshold gate can be considered a generalized C-element. This gate, namely  $m$ -of- $n$  threshold gate with hysteresis, and denoted as TH $m$  $n$ , has  $n$  input signals (wires) and one output, shown in Fig. 3.

This gate has two important properties, threshold behavior and hysteresis behavior. The threshold behavior requires that the output becomes 1 if at least  $m$  of the  $n$  inputs has become 1. The hysteresis behavior requires that the output only changes after a sufficiently complete set of input values have been established. In the case of a transition from 0 to 1, the output remains at 0 until at least  $m$  of the  $n$  inputs become 1. In the case of a transition 1 to 0, the output remains at 0 until all  $n$  inputs become 0. As special examples, an  $n$ -of- $n$  gate is an  $n$ -input Muller C-element, while a 1-of- $n$  gate corresponds to an  $n$ -input OR gate. Transistor-level design for  $m$ -of- $n$  threshold gates is described in [11]. As an example, a 2-of-3 gate having inputs A, B, and C is shown in Fig. 4. It includes four blocks (Go to 0, Go to 1, Hold 0, Hold 1), one inverter, and two feedback transistors (pMOS for Hold 0 and nMOS for Hold 1).

In order to optimize NCL circuits in terms of propagation delay, gate count, and transistor count, weighted threshold gates are proposed and denoted as TH $m$  $n$ W $w_1, w_2, \dots, w_r$ ,  $r < n$ , where  $n$  is the number of inputs,  $m$  is the gate's threshold, and  $w_1, w_2, \dots, w_r$  are the weights of input 1, input 2, ..., input  $r$ , respectively. Similarly, a TH $m$  $n$ W $w_1, w_2, \dots, w_r$  operation

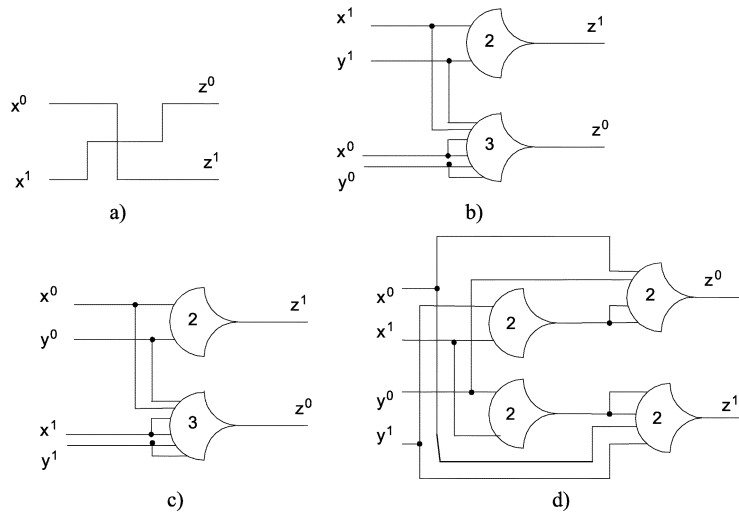


Fig. 6. NCL implementations of basic logic gates: (a) INVERT; (b) AND gate; (c) OR gate; and (d) XOR gate.

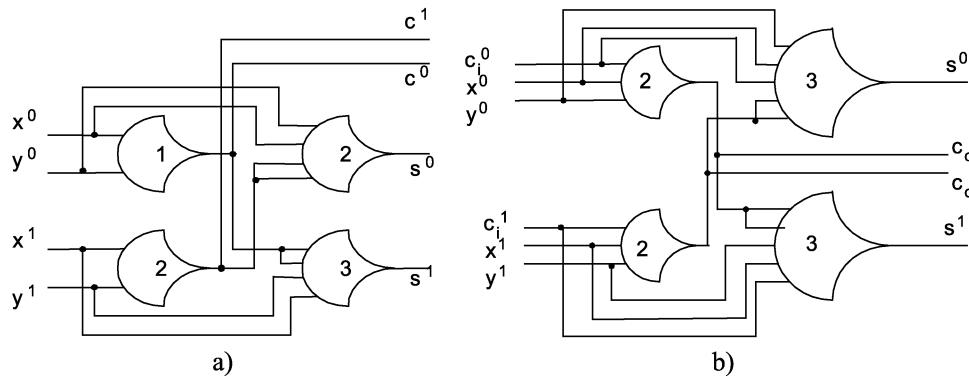


Fig. 7. Optimized NCL half adder and full adder. (a) NCL half adder. (b) NCL full adder.

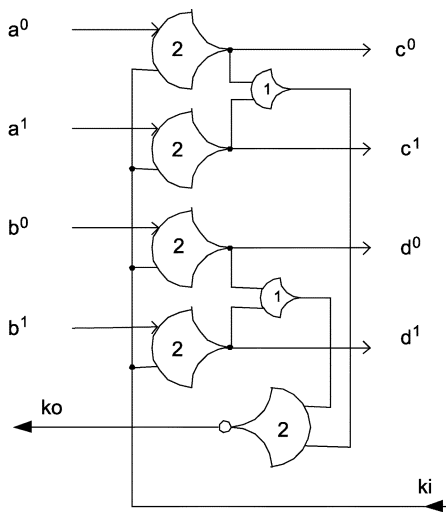


Fig. 8. Illustration of 2-b NCL register.

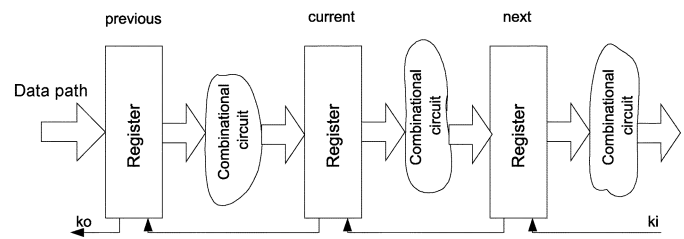


Fig. 9. Basic NCL pipeline structure.

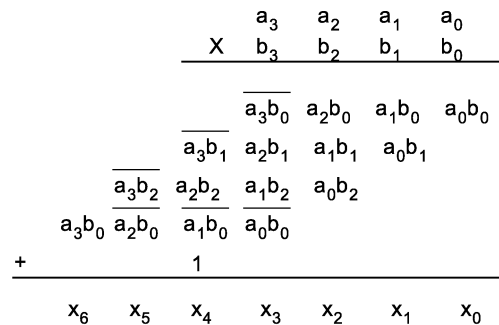


Fig. 10.  $4 \times 4$ -b multiplication.

can be described as follows: the output remains at 1 until all  $n$  inputs become 0 (same as TH $mn$ ); the output remains at 0 until the threshold algebra

$$(W_1 \times A_1) + (W_2 \times A_2) + \dots + (W_r \times A_r) + A_{r+1} + A_{r+2} + \dots + A_n \geq m$$

where  $A_1, A_2, \dots, A_n$  are the  $mn$  inputs to the gate. The schematic of weighted threshold gate TH23W2 is shown in Fig. 5.

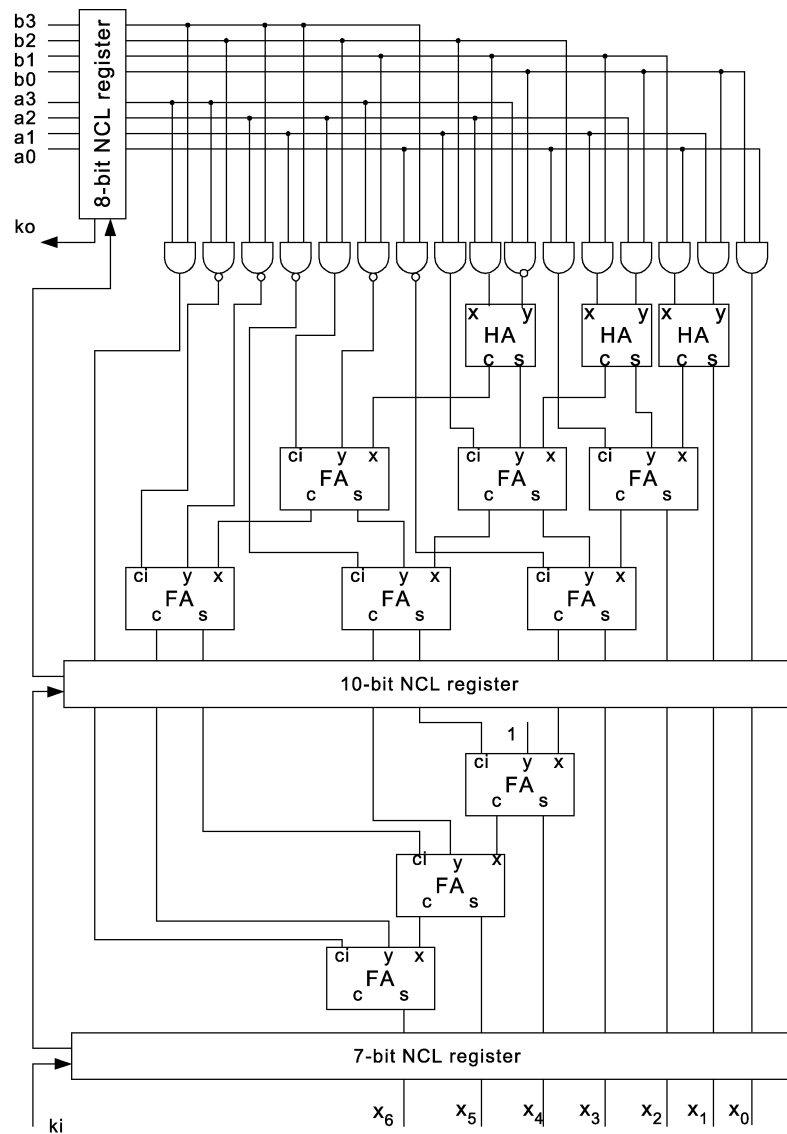


Fig. 11. Two-stage pipelined NCL  $4 \times 4$  multiplier.

#### D. Combinational Circuits and Registers

With the threshold gates, one can build NCL circuits that are speed independent according to a certain rule. Compared with traditional CBL, NCL usually requires more transistors when implementing dual-rail logic functions that satisfy the completeness of input criterion except an inverter. Fig. 6 shows the NCL implementations of the logic function of INVERT, AND, OR, and XOR, respectively. Optimized NCL half adder and full adder are illustrated in Fig. 7.

In fact, the presence of NULL is used as a time reference in NCL circuits. The consecutive DATA input to a combinational circuit should be separated by NULL in the time domain. DATA and NULL pass the combinational circuit alternatively.

To build a system out of NCL combinational logic circuits, the student must be able to manage the communication and interaction among its component circuits. There must be an asynchronous register, as shown in Fig. 8, which monitors completeness of resolution as well as readiness to accept new input data sets and stores the complete set of data values and the all NULL

values between circuits. Each register gate (TH22) receives one wire with a data meaning from the data path and one wire with a control meaning. The TH22 (detection gate) with an inverted output is a gate to detect the completion of the register output.

Fig. 9 shows a basic NCL pipeline structure. The input request for each register gate comes from the detection gate of the next register. Assume that all the circuits are in a NULL state and that the current detection gate and the next detection gate is requesting a DATA wave front and that the previous register is presenting a complete DATA set to its combinational circuit. As the wave front propagates through the previous circuit to the current register, the current register passes the data since its control line is DATA. When a complete data set is recognized by the current detection gate, the current detection gate transitions its control line to the previous register to NULL to indicate that it has received and stored the data wave front and that the previous register can pass a NULL wave front. The requested NULL wave front from the previous register can arrive at the current register but, as long as its control line is DATA, the NULL wave front will be blocked, and the current register will

```

library SYNOPSIS;
use SYNOPSIS.ATTRIBUTES.ALL;

library ieee ;
use ieee.std_logic_1164.all ;

PACKAGE ncl_logic IS

TYPE ncl_ulogic IS ( 'U', -- Uninitialized
                    'X', -- Forcing Unknown
                    '0', -- Forcing 0
                    '1', -- Forcing 1
                    'N', -- NULL
                    'Z', -- High Impedance
                    '-' -- Don't care
                  );
SUBTYPE ncl_logic IS resolved ncl_ulogic;
TYPE ncl_logic_vector IS ARRAY ( NATURAL RANGE (<>) ) OF ncl_logic;

.....

procedure hysteresis( ain : in ncl_logic_vector; -- vector / vector
                    signal zout : inout ncl_logic_vector );

.....

end ncl_logic;

```

Fig. 12. Declarations of NCL\_LOGIC type and hysteresis procedure in “ncl\_logic” package.

maintain the presentation of the set of DATA values to the current circuit. The control line for the current register will remain DATA until the DATA wave front has propagated through the current circuit and has been received by the next register. When the next register receives and stores the DATA wave front, the DATA set no longer needs to be maintained by the current register. The next detection gate detects the complete DATA set and transitions its acknowledge line to NULL to indicate that it has received the DATA wave front and that the current register can allow a NULL wave front through.

#### E. Design Example—4 × 4 Multiplier

An NCL 4 × 4 Baugh–Wooley multiplier can be constructed by using NCL AND gates, half adders, full adders, and registers. The multiplication chart for computing the output  $x_n$ ,  $n = 0, 1, 2, 3, 4, 5, 6$ , is shown in Fig. 10, where  $a_3a_2a_1a_0$  and  $b_3b_2b_1b_0$  are multiplicands,  $x_6x_5x_4x_3x_2x_1x_0$  is product, and  $a_3$ ,  $b_3$ , and  $x_6$  are sign bits of multiplicands and product.

An effective way to improve data throughput is to place registers on a feedforward cut set of the data path to split the combinational circuit into several stages. As an example, a two-stage pipelined NCL 4 × 4 multiplier is shown in Fig. 11, where a 10-b register splits the whole multiplier into two stages.

#### F. Very-High-Speed Integration Circuit Hardware Description Language Simulation at Register Transfer Level

NCL circuits can be designed using very-high-speed integration circuit hardware description language (VHDL) based on commercial high-level design tools with minor adjustments to account for the asynchronous behavior. A three-valued logic type with values {“Null,” “1,” “0”} should be introduced. Hysteresis behavior should be described inside combinational processes, though it is ignored during synthesis. These Null/DATA behavior and hysteresis are modeled by a package “ncl\_logic”.

```

procedure hysteresis( ain : in ncl_logic_vector;
                    signal zout : inout ncl_logic_vector )
is
begin

    if is_NULL( ain ) then
        zout <= set_NULL( zout ); -- set to null
    elsif is_data( ain ) then
        null; -- if data, allow change
    else
        zout <= zout; -- if 'X' makes no change
    end if;
end hysteresis;

```

Fig. 13. Definition of hysteresis procedure.

```

library ncl;
use work.ncl_logic.all;
entity multiplex
Port ( a, b, s : in ncl_logic;
      z : out ncl_logic);

architecture ncl of multiplex is

begin

    process(a,b,s)
    begin
        if s = '1' then
            z <= a;
        else
            z <= b;
        end if;
        hysteresis(a,b,x,z);
    end process;
end ncl;

```

Fig. 14. VHDL code for an NCL MUX.

TABLE II  
COURSE OUTLINE

I. Introduction
1) Why asynchronous design?
2) Category of asynchronous circuits
II. Basic Principles of Asynchronous Circuit Design
1) Communication protocols
2) Huffman circuits
3) Speed independent circuits
4) NCL design methodology
III. Arithmetic Circuit Design
1) Adders: carry propagation adder, look ahead adder
2) Multiplexor
3) Multipliers: Bough Wooley, Booth encode, Wallace tree
IV. Simulation of Asynchronous Digital Circuits
1) Basic VHDL and Synopsys tools
2) Asynchronous circuit simulation
3) NCL packages: ncl_logic, ncl_arith, ncl_signed, ncl_unsigned
4) Simulation using Synopsys
V. Applications
1) FIR filter design
2) Asynchronous rings and divider
3) Low power issues in asynchronous circuits

In addition, packages “ncl\_arith”, “ncl\_signed”, and “ncl\_unsigned” are used to support NCL arithmetic operations.

In the “ncl\_logic” package, a type NCL\_LOGIC is defined, shown in Fig. 12. Based on this type, all basic Boolean functions (AND, NAND, OR, NOR, etc.) as well as convention functions defined in IEEE 1164’s “std\_logic” package are overloaded for NCL\_LOGIC and NCL\_LOGIC\_VECTOR.

Hysteresis behavior is described by a predefined procedure in the "ncl\_logic" package shown in Fig. 13. This procedure is called at the end of every process with all process signals as parameters. The procedure hysteresis() assures that the output is modified only when the inputs are either all NULL or all DATA.

As an example, the student can describe an NCL MUX in VHDL, as shown in Fig. 14.

### G. Course Outline

Table II lists the topics that are covered in a one-semester course format. The introduction presents the advantages of asynchronous circuits and their categories. After introduction several lectures are devoted to the fundamental principles of asynchronous circuit design. Typical arithmetic circuits presented include adders, multiplexor, and multipliers. VHDL is introduced as a tool to simulate asynchronous circuits. Some packages are explained to help students understand how to use VHDL for NCL circuit simulation. The course concludes with some applications.

## IV. CONCLUSION

This paper introduces a means of teaching asynchronous digital circuit design in an electrical and computer engineering curriculum. The authors present the design flow for NCL to give students a basic understanding of asynchronous circuit. The proposed design methodology is straightforward, and students can design a specific asynchronous circuit without complicated background theory. For those students who want to participate in advanced asynchronous research, however, they need to further their reading.

## ACKNOWLEDGMENT

The authors would like to thank Dr. M. Hagedorn at Theseus Logic, Inc., for the discussion on the Null Convention Logic (NCL).

## REFERENCES

- [1] S. B. Furber, J. D. Garside, P. Riocreux, S. Temple, P. Day, J. Liu, and N. C. Paver, "AMULET2e: An asynchronous embedded controller," *Proc. IEEE*, vol. 87, pp. 243–256, Feb. 1999.
- [2] A. J. Martin, A. Lines, R. Manohar, M. Nystroem, P. Penzes, R. Southworth, and U. Cummings, "The design of an asynchronous MIPS R3000 microprocessor," in *Adv. Res. VLSI*, Sept. 1997, pp. 164–181.
- [3] H. Van Gageldonk, D. Baumann, K. Van Berkel, D. Gloor, A. Peeters, and G. Stegmann, "An asynchronous low-power 80c51 microcontroller," in *Proc. Int. Symp. Advanced Research Asynchronous Circuits Systems*, 1998, pp. 96–107.

- [4] K. V. Berkel, R. Burgess, J. Kessels, A. Peetrs, M. Roncken, and F. Schalij, "A fully-asynchronous low-power error corrector for the DCC player," in *IEEE Int. Solid-State Circuits Conf.*, Feb. 1994, pp. 88–89.
- [5] S. Hauck, "Asynchronous design methodologies: An overview," *Proc. IEEE*, vol. 83, pp. 69–93, Jan. 1995.
- [6] K. M. Fant and S. A. Brandt, "NULL Convention Logic: A complete and consistent logic for asynchronous digit circuit synthesis," in *Int. Conf. Application Specific Systems, Architectures, Processors*, 1996, pp. 261–273.
- [7] I. E. Sutherland and J. Ebergen, "Computers without clocks," *Sci. Amer.*, pp. 62–69, Aug. 2002.
- [8] M. Renaudin and B. El Hassan, "The design of fast asynchronous adder structures and their implementation using DCVS logic," in *1994 IEEE Int. Symp. Circuits Systems*, 1994, pp. 291–294.
- [9] C. L. Seitz, "System timing," in *Introduction to VLSI Systems*: Addison-Wesley, 1980, pp. 218–262.
- [10] K. van Berkel, "Beware the isochronic fork," *Integr. VLSI J.*, vol. 13, no. 2, pp. 103–128, June 1992.
- [11] G. E. Sobelman and K. M. Fant, "CMOS circuit design of threshold gates with hysteresis," in *Proc. 1998 IEEE Int. Symp. Circuits Systems*, 1998, pp. 61–64.

**Jiann S. Yuan** (S'85–M'88–SM'02) received both the M.S. and Ph.D. degrees from the University of Florida, Gainesville, in 1984 and 1988, respectively.

He joined the faculty at the University of Central Florida (UCF), Orlando, in 1990, after one year of industrial experience at Texas Instruments, Inc., where he was involved with the 16-MB CMOS DRAM design. Currently, he is a Professor and Director of the Chip Design and Reliability Laboratory at the School of Electrical Engineering and Computer Science at UCF. He has published more 200 papers in refereed journals and conference proceedings, and he has authored the books *Semiconductor Device Physics and Simulation* (New York: Plenum, 1998) and *SiGe, GaAs, and InP Heterojunction Bipolar Transistors* (New York: Wiley, 1999).

Dr. Yuan is a Member of Eta Kappa Nu and Tau Beta Pi. He has received many awards, including the Distinguished Researcher Award from UCF (1993, 1996, and 2002), the Outstanding Engineering Educator Award, the IEEE Orlando Section and Florida Council (1993), and the Teaching Incentive Program (TIP) award from UCF (1995). He is listed in *Who's Who in American Education* and *Who's Who in Science and Engineering*. He serves regularly as a reviewer for the IEEE TRANSACTIONS ON ELECTRON DEVICES, IEEE ELECTRON DEVICE LETTERS, and SOLID-STATE ELECTRONICS. He is an editor of the IEEE TRANSACTIONS ON DEVICE AND MATERIALS RELIABILITY.

**Weidong Kuang** received the B.S. and M.S. degrees from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991 and 1994, respectively. He is currently working toward the Ph.D. degree at the University of Central Florida, Orlando.

From 1994 to 1999, he was with the Advanced Radar Laboratory in No. 23 Institute of China Aerospace Corporation, Beijing, China, where he worked on phased-array radar design and development. His research interests are in radar signal processing, very large system integration (VLSI) low-power architecture for digital signal processing, and asynchronous circuit design.