

## XVII. DIGITAL SYSTEM DESIGN AUTOMATION\*

### Academic and Research Staff

Prof. F. F. Lee  
F. C. Hsieh  
J. B. Kasarda

### Graduate Students

B. S. Franklin  
H-m. D. Toong

#### A. MACHINE-AIDED DIGITAL SYSTEM DESIGN LANGUAGE (DSL/1)

In the development of a computer-oriented language, it is often helpful to describe the language in some standard form. We present the syntax of the Digital Specification Language (DSL/1) in Backus-Naur form. See Table XVII-1.

A brief description of each of the items will help to clarify some requirements of the language that are not easily specified in the syntax. The basic entities of the language will be described.

1. Registers: A register corresponds to a physical storage unit within the subsystem, or machine. In the declaration, the size (number of bits) is specified.

2. Subregisters: A subsection of a register may be used often enough to warrant the use of another name to refer to the subsection. In the subregister declaration, the parent register name is given, together with the start and end bits, which automatically defines the subregister size.

3. Concatenated registers: When the designer wishes to refer to several register sections or registers as one unit he may declare a name for the entire unit. The subsections are concatenated left to right, with the least significant bit of the left-hand register just to the left of the most significant bit of the right-hand register.

4. Booleans: This is a special case of a register, namely a one-bit value that may be viewed as a logical quantity (TRUE or FALSE), or as a one-bit storage value (1 or 0).

5. Instants: The instant is the basic timing element. It has zero duration, and divides the time region into two parts, the "before" region, and the "after" region. Instants are separated from other instants by an interval of time.

6. Transitions: A method of detecting changes of state within a machine and then actuating another action is based on a transition. The detection of the change is the transition and, like an instant, it divides the time region into "before" and "after" regions.

---

\*This work was supported by the Joint Services Electronics Programs (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract DA 28-043-AMC-02536(E).

Table XVII-1. DSL SYNTAX

```

<DESCRIPTION> ::= <SUBSYSTEM>|<DESCRIPTION><SUBSYSTEM>

<SUBSYSTEM> ::=  SUBSYSTEM <SUBSYSTEM IDENTIFIER>†<SUBSYSTEM BODY> SYSEND

<SUBSYSTEM IDENTIFIER> ::= <NAME>

<SUBSYSTEM BODY> ::= <DECLARATION> | <SUBSYSTEM BODY><SUBSYSTEM BODY> |
                   <SUBSYSTEM BODY><SPECIFICATION BLOCK>

<DECLARATION> ::= <GLOBAL REGISTER DECLARATION> $ |
                 <REGISTER DECLARATION> $ |
                 <SUBREGISTER DECLARATION> $ |
                 <CONCATENATED REGISTER DECLARATION> $ |
                 <GLOBAL INSTANT DECLARATION> $ |
                 <INSTANT DECLARATION> $ |
                 <TRANSITION DECLARATION> $ |
                 <MEMORY DECLARATION> $ |
                 <GLOBAL PCCLEAN DECLARATION> $ |
                 <PCLEAN DECLARATION> $ |
                 <GLOBAL DEFINE> $ |
                 <DEFINE> $ |
                 <COMMENT>

<MEMORY DECLARATION> ::= MEMORY <MEMORY STRING> |
                    <MEMORY DECLARATION>,<MEMORY STRING>

<MEMORY STRING> ::= <MEMORY IDENTIFIER>(<INTEGER>,<INTEGER>)
                  <MEMORY ADDRESS REGISTER IDENTIFIER>

```

<MEMORY IDENTIFIER> ::= <NAME>

<MEMORY ADDRESS REGISTER IDENTIFIER> ::= <REGISTER IDENTIFIER>

<REGISTER IDENTIFIER> ::= <NAME>

<GLOBAL REGISTER DECLARATION> ::= GLOBAL <REGISTER DECLARATION>

<REGISTER DECLARATION> ::= REGISTER <REGISTER IDENTIFIER> (<INTEGER>) |  
 <REGISTER DECLARATION>, <REGISTER IDENTIFIER> (<INTEGER>)

<NAME> ::= <ALPHABET> | <NAME> <ALPHABET> | <NAME> <DIGIT>

<ALPHABET> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<DIGIT> ::= <OCTAL DIGIT> | 8 | 9

<OCTAL DIGIT> ::= 1|2|3|4|5|6|7|0

<INTEGER> ::= <DECIMAL CONSTANT> | <OCTAL CONSTANT>

<DECIMAL CONSTANT> ::= <DECIMAL STRING> | <DECIMAL STRING> D

<DECIMAL STRING> ::= <DIGIT> | <DECIMAL STRING> <DIGIT>

<OCTAL CONSTANT> ::= <OCTAL STRING> B

<OCTAL STRING> ::= <OCTAL DIGIT> | <OCTAL STRING> <OCTAL DIGIT>

```

<SUPERREGISTER DECLARATION> ::= SREGISTER <REGISTER IDENTIFIER>=
    <REGISTER ITEM>|
    <SUBREGISTER DECLARATION>,<REGISTER IDENTIFIER>=<REGISTER ITEM>

<REGISTER ITEM> ::= <REGISTER IDENTIFIER>|<REGISTER IDENTIFIER>(<INTEGER>)|
    <REGISTER IDENTIFIER>(<INTEGER>,<INTEGER>) | <BOOLEAN IDENTIFIER>

<CONCATENATED REGISTER DECLARATION> ::=
    CREGISTER <REGISTER IDENTIFIER>=<REGISTER STRING>|
    <CONCATENATED REGISTER DECLARATION>,<REGISTER IDENTIFIER>=
        <REGISTER STRING>

<REGISTER STRING> ::= <REGISTER ITEM>|<REGISTER STRING>'<REGISTER STRING>

<GLOBAL INSTANT DECLARATION> ::= GLOBAL <INSTANT DECLARATION>

<INSTANT DECLARATION> ::= INSTANT <INSTANT IDENTIFIER>|<INSTANT DECLARATION>,<
    INSTANT IDENTIFIER>

<GLOBAL BOOLEAN DECLARATION> ::= GLOBAL <BOOLEAN DECLARATION>

<BOOLEAN DECLARATION> ::= BOOLEAN <BOOLEAN IDENTIFIER>|
    <BOOLEAN DECLARATION>,<BOOLEAN IDENTIFIER>

<BOOLEAN IDENTIFIER> ::= <NAME>

<BOOLEAN ITEM> ::= <BOOLEAN IDENTIFIER>|<REGISTER IDENTIFIER>(<INTEGER>)

<GLOBAL DEFINE> ::= GLOBAL <DEFINE>

```

```

<DEFINE> ::= DEFINE <DTYPE> <DBLOCK>

<DTYPE> ::= REGISTER | ECLEAN | INSTANT

<DBLOCK> ::= <REGISTER IDENTIFIER> = <REGISTER EXPRESSION> |
             <ECLEAN IDENTIFIER> = <ECLEAN EXPRESSION> |
             <INSTANT IDENTIFIER> = <INSTANT EXPRESSION> |
             <TBLOCK>,<DBLOCK>

<INSTANT IDENTIFIER> ::= <NAME>

<SPECIFICATION BLOCK> ::= <SPECIFICATION BLOCK BODY> $

<SPECIFICATION BLOCK BODY> ::= AT <INSTANT EXPRESSION> DO <ASSIGNMENT BODY> |
             IF <ECLEAN EXPRESSION> THEN <SPECIFICATION BLOCK BODY> IFEND |
             IF <ECLEAN EXPRESSION> THEN <SPECIFICATION BLOCK BODY> ELSE
             <SPECIFICATION BLOCK BODY> IFEND

<INSTANT EXPRESSION> ::= <INSTANT IDENTIFIER> | <TRANSITION IDENTIFIER> |
             <TRANSITION> | <INSTANT EXPRESSION> AND <ECLEAN EXPRESSION> |
             <ECLEAN EXPRESSION> AND <INSTANT EXPRESSION> |
             <INSTANT EXPRESSION> CB <INSTANT EXPRESSION> |
             (<INSTANT EXPRESSION>) | <INSTANT EXPRESSION> + <INTERVAL>

<TRANSITION> ::= <ECLEAN EXPRESSION> ? <BOOLEAN CONSTANT> |
             <ECLEAN EXPRESSION> => <ECLEAN CONSTANT>

<TRANSITION DECLARATION> ::= TRANSITION <TRANSITION IDENTIFIER>=
             (<TRANSITION>) |
             <TRANSITION DECLARATION>,<TRANSITION IDENTIFIER>=
             (<TRANSITION>)

```

<TRANSITION IDENTIFIER> ::= <NAME>

<ECOLEAN EXPRESSION> ::= <ECOLEAN CONSTANT> | <ECOLEAN ITEM> |  
 <REGISTER EXPRESSION> <RELATIONAL OPERATOR> <REGISTER EXPRESSION> |  
 <ECCLEAN EXPRESSION> <ECCLEAN OPERATOR> <ECOLEAN EXPRESSION> |  
 NOT <ECOLEAN EXPRESSION> | ~<ECOLEAN EXPRESSION> |  
 (<ECCLEAN EXPRESSION>)

<ECCLEAN CONSTANT> ::= TRUE | FALSE | 1 | 0

<RELATIONAL OPERATOR> ::= EQ | NE | GT | LT | GE | LE | = | <> | > | < | <= | >=

<BOCLEAN OPERATOR> ::= AND | OR | XOR | .

<ASSIGNMENT BODY> ::= <TRANSFER> | <INSTANT ACTIVATION> |  
 IF <ECOLEAN EXPRESSION> THEN <ASSIGNMENT BODY> IFEND |  
 IF <ECOLEAN EXPRESSION> THEN <ASSIGNMENT BODY> ELSE  
 <ASSIGNMENT BODY> IFEND |  
 <ASSIGNMENT BODY>, <ASSIGNMENT BODY>

<TRANSFER> ::= <REGISTER STRING> = <REGISTER EXPRESSION> |  
 <ECCLEAN ITEM> = <ECOLEAN EXPRESSION>

<INSTANT ACTIVATION> ::= <INSTANT IDENTIFIER> = \* + <INTERVAL>

<INTERVAL> ::= <REAL CONSTANT>

<REAL CONSTANT> ::= <DECIMAL CONSTANT> . |  
 <DECIMAL CONSTANT> . <DECIMAL CONSTANT> | <DECIMAL CONSTANT>

```

<REGISTER EXPRESSION> ::= <REGISTER STRING> | <REGISTER CONSTANT> |
    <ECCLEAN EXPRESSION> | NOT <REGISTER EXPRESSION> |
    ~ <REGISTER EXPRESSION> |
    <REGISTER EXPRESSION> <LOGICAL OPERATOR> <REGISTER EXPRESSION> |
    <REGISTER EXPRESSION> <ARITHMETIC OPERATOR> <REGISTER EXPRESSION> |
    (<REGISTER EXPRESSION>)

```

```

<REGISTER CONSTANT> ::= <INTEGER>

```

```

<LOGICAL OPERATOR> ::= AND | OR | XOR | .

```

```

<ARITHMETIC OPERATOR> ::= ADD | SUB | + | -

```

```

<CHARACTER> ::= <ALPHABET> | <DIGIT> | <TERMINATOR>

```

```

<TERMINATOR> ::= = | < | ; | : | % | ' | > | * | ( | ) | + | & | ! | " | ? | / | - | . | , | $

```

7. Memory: A memory system is a collection of equal-size registers grouped together under a single name, and for which only one element may be referenced at any one time. The addressing scheme is accomplished through a memory address register, whose declaration appears in a register declaration statement.

8. Specification block: This group of statements specifies the actual operation of the machine. Its component parts are the following.

(a) The instant activation phrase that determines at what time this block is to be executed.

(b) The Boolean condition that specifies what state the system must be in to enable execution.

(c) Assignment body containing the transfer-path specifications and generation of new instants.

(XVII. DIGITAL SYSTEM DESIGN AUTOMATION)

9. Globals: To allow the communication of data and machine states to the external observer, the designer specifies what registers, booleans and instants are to appear at the interface. If a register is contained within subsystem A and is used in subsystem B, it is declared internal global to A, and external global to B.

10. Subsystem: The set of declarations and specification blocks defining the operation of a machine is termed a subsystem, and is given a name by the designer.

11. Description: A description is a collection of subsystems that, when taken as an entire group, performs a macrofunction. An example would be a computer description, as opposed to an arithmetic subsystem.

Further information about the implementation of a modified subset of DSL/1 may be obtained from a thesis by John B. Kasarda.<sup>1</sup> This thesis describes a subset of the DSL/1 presented here, the compiler and simulation system, and gives an example of output produced by the system.

F. F. Lee

References

1. J. B. Kasarda, "A Machine-Aided Digital Systems Design Language," S. B. Thesis, Department of Electrical Engineering, M. I. T., February 1970.