1-1-1990

A study of children learning multicolumn addition with microcomputer software support.

Hyman S. Edelstein
University of Massachusetts Amherst

A STUDY OF CHILDREN
LEARNING MULTICOLUMN ADDITION
WITH MICROCOMPUTER SOFTWARE SUPPORT

A Dissertation Presented

by

HYMAN S. EDELSTEIN

Submitted to the Graduate School of the
University of Massachsetts in partial fulfillment
of the requirements for the degree of

DOCTOR OF EDUCATION

February 1990

School of Education

A STUDY OF CHILDREN
LEARNING MULTICOLUMN ADDITION
WITH MICROCOMPUTER SOFTWARE SUPPORT

A Dissertation Presented

by

HYMAN S. EDELSTEIN

Approved as to style and content by:

Howard A. Peelle

Howard A. Peelle, Chairperson of Committee

George Forman

George Forman, Member

Alexander Pollatsek

Alexander Pollatsek, Member

Marilyn Haring-Hidore, Dean
School of Education

ACKNOWLEDGEMENTS

I wish to acknowledge my deep appreciation and thanks:

To Howard A. Peelle for his unwavering encouragement, help, and advice throughout this long course of graduate study and research. Our many discussions were always spirited and invariably provided me with renewed motivation and sense of direction.

To George Forman and Alexander Pollatsek for their many helpful suggestions and perceptive critiques of this work.

To Michael Greenebaum and Joan Langley of Marks Meadow School and to Donald Gasiorowski, Barbara Filmore, and Sherry Gelinas of the Morgan School for their enthusiastic cooperation and support.

To Richard Konicek, Roger and Barbara Miller, Merton and Hilda Kahne, and to Steffanie Schamess for their inspiration and encouragement.

To my family, Sally, Carol & Andrew, Arthur, Alan & Lisa, for their long suffering support, patience, and love.

And to all those wonderful children who willingly participated and who I sincerely hope will benefit from this research effort.

ABSTRACT

A STUDY OF CHILDREN
LEARNING MULTICOLUMN ADDITION
WITH MICROCOMPUTER SOFTWARE SUPPORT

FEBRUARY 1990

HYMAN S. EDELSTEIN, B. S., UNIVERSITY OF MASSACHUSETTS

Ed.D., UNIVERSITY OF MASSACHUSETTS

Directed by: Professor Howard A. Peelle

Three computer-aided tutoring procedures were devised to teach multicolumn addition according to the standard school algorithm, one procedure to each of three groups of 2nd-grade children. The key differences between groups were the demands placed on short term memory and the amount of conceptual understanding the procedures attempted to teach. Each child solved a sequence of two-digit problems on a computer screen by touching each digit with a light pen in the correct sequence.

The control group did not receive on-screen number-fact assistance. One treatment ("assisted") group did receive on-screen number-fact assistance, testing the hypothesis that the algorithm is learned more effectively when learned first as a sequence of procedural steps alone, without subjects' need to recall number-facts. A second treatment ("simulation") group received the same on-screen assistance along with an additional display of simulated blocks which, like concrete manipulative materials, represented symbol manipulations. The simulation group tested a second hypothesis that a concurrent display of the meaning of procedural steps contributes to even more effective algorithmic learning.

T-tests (one-tailed, 5% level) applied pair-wise to pretest/posttest difference scores indicated support for the first hypothesis but not for the second, an indication that 2nd-grade children learn the addition algorithm more effectively if demand on short term memory is temporarily lifted.

A descriptive framework called "superposition of frames" is proposed to account for anomalies in findings and for the rich diversity of errors generally manifested by children in multidigit addition. Drawing on current concepts in cognitive psychology and mathematics education, this description suggests that children's mathematical knowledge is fragmented into isolated, unstable, and sometimes entrenched frames of knowledge. When a child finds appropriate correspondences between frames and initiates a superposition of frames, the child's procedural and conceptual knowledge, previously in disarray, may then become integrated. Implications for elementary mathematics instruction are discussed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

Algorithm

A rule-driven, computational procedure which automatically generates a desired transformation of a mathematical expression or achieves a solution to a problem.

Algorithmic behavior

The act of implementing or performing an algorithm, which involves recall of the procedural schema, that is, the rules and proper sequencing of the steps in the procedure; perception of the symbols, their arrangements and transformations, implying some sort of perceptual organization or gestalt; and motor activity in physically manipulating or recording the symbols being processed.

Light pen

An input/output device in the shape of a conventional pen, which allows direct communication between the subject and a computer screen, by-passing the keyboard. When the tip of the pen is held against the screen, a photocell embedded in the tip detects the passage of the cathode ray beam scanning across the inside face of the cathode ray tube. The scanning process is precisely timed; consequently, the position of the pen tip can be determined by software calculation of what point in time in the scanning cycle the passage of the beam is detected by the pen's photocell.

Mapping instruction

A term used by Resnick & Omanson (1987): instruction that requires the learner to perform the algorithm both with manipulative blocks and in writing, "maintaining a step-by-step correspondence between the blocks and written symbols throughout the problem ... designed to help children link their knowledge of the principles [of regrouping, place value, etc.] to written subtraction". In this study, mapping instruction takes the form of showing or representing the quantities and their manipulations as simulated blocks on the screen, rather than requiring physical manipulations of blocks or writing.

Prohibition learning

A term used by Resnick & Omanson (1987): learning that occurs by practicing an algorithm under conditions in which no incorrect steps are permitted; that is, the instructor cautions the learner whenever an incorrect move is made. The way the light pen is used in the proposed study is a form of prohibition instruction but with minimal intervention by the instructor. The subject can progress through the algorithm only by making the correct moves; incorrect moves elicit a quiet beep.

CHAPTER 1

INTRODUCTION

Among the many difficulties elementary school children encounter in arithmetic is learning to master basic algorithms such as multicolumn addition, where children are required to process numeric symbols according to well-defined procedural schemes. These tasks are generally regarded as relatively easy mechanical manipulations of symbols, a low level rote skill which can be performed with little understanding of mathematical principles (Davis, 1988; Stein, 1988). The superficial ease with which an algorithm may ultimately be performed masks both the automatization laboriously achieved over a long period of practice and the complexity of concepts and subroutines underpinning the algorithm.

For a novice, difficulties of learning an algorithm should not be so lightly dismissed. The memory demand is not inconsiderable. The child needs to recall number facts, sequences of operations, conditions triggering particular sequences of operations, proper placement of the numerals generated by the algorithm as well as various explanations, purposes, and meanings for all these, with or without full understanding. To embark on the long complex undertaking of doing mathematics, even at an elementary level, is to endure countless little failures, and perhaps in the end, overwhelming failure. Success is beset in general with misconceptions, imperfectly remembered and inappropriately applied operations, disorientations, and in particular with lack of fundamental understanding about place value, lapses of memory for number facts, and the distractions of having to reconstruct number facts by counting. A child learning the symbol manipulations of an algorithm and their meanings is facing a significant cognitive challenge.

Performing an algorithm would seem to be a rote exercise of procedural knowledge requiring little conceptual knowledge. A closer examination of the process reveals other cognitive processes at work such as perceptual organization, concept formation, and planning. Total cognitive demands may very well exceed the student's capacity in a first encounter with a new algorithm. This raises a number of general questions:

How successfully does a student manage limited processing capacity while assimilating a new algorithm? Would an algorithm be learned more effectively if some of the demand on short term memory (STM) capacity were lifted temporarily? Or put another way, would an algorithm separated into distinct but parallel sub-processes, each of which is learned separately and automated to some degree, then combined, result in more effective learning?

May understanding and the ability to manipulate symbols proceed independently of one another, at least for a time? To what extent does having an understanding of mathematical principles facilitate learning an algorithm? Conversely, does the learning of an algorithm facilitate understanding mathematical principles?

What aspects of an algorithm tend to emerge as buggy procedures?

In order to address these questions in this study, computer software was designed that permitted a student to learn the procedural steps of multicolumn addition without having to recall number facts. The intention here is to alleviate possible short term memory overload at the initial stages of learning an algorithm. A treatment group having this kind of software support was compared with a control group which used the same software but was required to recall number facts initially.

One hypothesis to be tested is that multicolumn addition is learned more effectively when learned first as a sequence of procedural steps alone and without initial recall of number facts than when the algorithm is learned along with required recall of number facts.

To address the question of the role of understanding mathematical principles in algorithmic performance, a second group was treated with an additional feature in the software. The second treatment group proceeded in the same way as the first treatment group but additionally saw displayed on the computer screen a concrete representation of the numbers (in the form of arrays of simulated blocks) as they were being manipulated.

A second hypothesis to be tested is that this simultaneous concrete display of the "meaning" of procedural steps contributes significantly to the effectiveness of learning the algorithm. Effectiveness is defined here as the fewest errors made with a maximum of understanding of the mathematical principles involved, as indicated in posttesting at the end of instruction.

A more general question may be raised about the necessity for studying multicolumn addition; after all, by the end of the third grade, most children have mastered addition. Multicolumn subtraction, on the other hand, continues to be difficult for many children throughout the elementary grades and has been the object of many studies. See particularly the seminal studies by Brown & Burton (1978) and Brown & VanLehn (1980). The reasons for the interest in subtraction are likely due to the relatively greater complexity of the subtraction algorithm. In contrast to subtraction, in multicolumn addition, digits in a column may be commuted; a carry does not change the value of any other digit; and zeros may be safely ignored. None of this applies to subtraction.

There are, however, several reasons why studies of multicolumn addition should be pursued:

First, it is typically the first formalism a child encounters that embodies the power of abstraction. Adding by recall of number-facts is limited to single digits; adding by counting-on becomes tedious, error-prone, and virtually impossible in practice when large multidigit numbers are to be added together. In multidigit addition, a child may acquire for the first time a sense that mathematics is a powerful tool.

Second, the relative success in learning the mechanics of the multidigit addition algorithm may obscure lack of understanding, especially place-value understanding, that may linger on, impeding a child's progress and generating hostile attitudes towards mathematics.

Third, there is only a fleeting time when addition bugs are as diverse and frequent as they are at the introduction of the algorithm. In this study of only 36 children, nearly 50 different kinds of errors in multicolumn addition were found, which provide a possibility of furthering our understanding of children's thinking processes.

And fourth, a study of multicolumn addition should contribute to helping resolve on-going debate in mathematics education over the relative emphasis placed on acquiring procedural knowledge or rote skills vs. conceptual knowledge or understanding of mathematical relationships.

An overriding reason for studying any aspect of elementary arithmetic at this time lies in the decline of children's mathematics performance in recent years reported by the National Assessment of Educational Progress (Kouba, Brown, Carpenter, et al., 1988) and by comparative studies indicating that American children are lagging behind children of other countries (Stevenson, Lee, & Stigler, 1986).

CHAPTER 2

REVIEW OF LITERATURE

An algorithm is defined by Suydam (1975) as "a method consisting of a finite number of steps taken in a preassigned order and reproducible, specifically adapted to the solution of problems of a particular category." The implication is that the method can be applied successfully without understanding. This review of literature begins with this major theme in mathematics education: rote skills versus understanding. It is followed by reviews of areas relevant to the hypotheses being tested in this study -- hypotheses concerning skill acquisition, memory research, educational studies of addition algorithms -- and ends with a review of the methodologies employed.

2.1 Skills and Understanding in Mathematics

Ever since Brownell (1935) called for learning with understanding in mathematics as a corrective to the prevailing rote associationist approaches of his day, mathematics education has been somewhat polarized between these two views. A study of algorithmic behavior would seem to fall into the camp of associationism, a psychological theory that was very influential in the early decades of this century. The theory justified the attention paid to learning algorithms and to drill and practice (Resnick & Ford, 1981). The teacher's task was to strengthen the bonds (stimulus-response chains) between the material to be learned (the stimulus) and the correct responses to the material presented. Thorndike's Law of Effect, a precursor of Skinner's principles of reinforcement, stated that when a response to a stimulus was rewarded, a "bond" or association was formed. The bond was strengthened by continued reward when the desired response continued to be made to the same stimulus. Thorndike's 1922 book, "The Psychology of Arithmetic" set forth the detailed bonds and habits which were needed to be formed if arithmetic were to be learned properly. The proper amount of practice was to be provided in the proper order in each class of problems.

Opposing these behaviorist methods, Brownell (1935) advocated instruction that stressed understanding of mathematical relationships: "One needs a fund of meanings, not a myriad of 'automatic responses'."

A study of any aspect of algorithmic learning would be significant if only for the fact that much of the time spent in elementary school mathematics is devoted to acquiring algorithmic skills. Such compulsive emphasis may result in piecemeal understanding of mathematics and an inability to solve problems other than textbook exercises (Carpenter, 1985). Rote learning alone is clearly inadequate for progress through mathematics (Hiebert & Lefevre, 1986), although it may be adequate for progress through a school system. Then it becomes a means of last resort by students who do not quite understand what they are doing but believe they are able to produce "correct" answers. Davis (1988) pronounces such behavior as ritualistic:

> Is mathematics really a matter of learning to perform a few meaningless rituals? .... What's wrong with teaching mathematics as a collection of meaningless rituals? .... students do deal with meanings; and when instructional programs fail to develop appropriate meanings, students create their own meanings which are sometimes not appropriate .... All of us use some rituals (or if you prefer, "procedures that we don't think about and may not understand") .... Is it inevitable that students will develop at first a ritual point of view? (Davis, 1988)

His concerns are amply supported in the literature. Morris (1981) found that a symptom of math anxiety is "memorization replaces understanding". Fremont (1971) described rote learning as one of the "time-honored enemies of effective mathematical learning". Allardice & Ginsburg, (1983): "Algorithms are learned in rote, meaningless ways and are easily forgotten .... Were the conceptual framework made available, then forgetfulness would be reduced." Stein (1988) paraphrases Gresham's law in economics (bad money drives good money out of circulation) for mathematics education, that "cultivation of algorithms replaces concern for thinking and writing":

> Algorithms of course are good and must be taught. .... But the temptation to emphasize drill over understanding is almost irresistible. It is much easier to teach the execution of an algorithm than the ability to analyze. Furthermore, an algorithm can be described in just a few minutes, and skill in its execution can be tested and scored easily. (Stein, 1988)

Executing an arithmetic algorithm correctly may be an end in itself, as in adding up a list of purchases, but generally it is employed in the larger context of solving a problem. As such it is seen to play a very important but supporting role as syntax (rules of symbol manipulation) to aspects of the problem that are charged with semantics (the meaning of the symbols). Romberg (1982) sees problem-solving as a semantic/syntactic process: first, comprehend the problem statement, then quantify the elements of the problem, express the semantics of the problem syntactically, carry out the procedural steps, and finally express the results of these operations. Dealing with syntax separately from semantics may lead to mere symbol manipulation without meaning. In the process of learning formal arithmetic procedures, many children stop analyzing problems and mechanically add and subtract without regard for the meaning or content of a problem (Carpenter & Moser, 1982).

Wearne & Hiebert (1988) offer a theory of how students become competent with the written symbols of the decimal fraction system. Symbols and rules take their meaning from real world referents but attain their power by becoming separated from these referents. Competence results from a cumulative and sequential mastery of four cognitive processes, two of them semantic and two syntactic:

<u>Semantic processes</u>

1. Connecting processes: learning to construct links between symbols and familiar referents, followed by:

2. Developing processes: learning procedures used to manipulate symbols, followed by:

<u>Syntactic processes</u>

3. Elaborating/routinizing process: learning to transfer syntax to other similar contexts by means of drill and practice and automating procedures.

4. Abstracting process: learning to construct a more abstract system on familiar rules and symbols.

Wearne & Hiebert conclude that it is preferable to develop meanings for symbols before practicing syntactic (algorithmic) routines.

Other researchers claim that a reverse sequence occurs in these processes. Without first automating procedures, without committing to memory commonly used facts and procedures, progress through mathematics may also be retarded (Gagne, 1983; Anderson, 1982, 1987). Even when algorithms are learned "with understanding", the understanding may be flawed. A student may misunderstand a procedure, or misapply a procedure that elsewhere is valid, and consistently introduce a "bug" (Brown & Burton, 1978).

In the 50 years since these debates began, cognitive psychological theories have become increasingly influential in mathematics instruction (Howson, Keitel, & Kilpatrick, 1981). More research is directed now towards how a student develops cognitively, or in information processing terms, how knowledge is represented, stored, and retrieved, as well as what metacognitive strategies and plans are used in problem-solving. Instructional programs are claimed to be more effective when they are designed around developing cognitive abilities and around the ways by which students construct their own knowledge. An effective instructional program would be defined as one by which a student not only acquires accurate algorithmic or computational skills quickly but also comes to understand the mathematical relationships required to solve problems beyond standard drill-and-practice exercises.

Recent expressions of these views are found in the literature on procedural and conceptual knowledge (Carpenter & Moser, 1982) and expert/novice problem-solving (Larkin, McDermott, Simon, & Simon, 1980). Mathematical procedures and concepts are not isolated skills and bits of knowledge but are related to other procedures and concepts. Capable students focus on this, the mathematical structure of a problem.

> Expert problem solvers tend to organize their knowledge in large related chunks on the basis of fundamental mathematical properties. Novices store their knowledge in more isolated bits or sort it on the basis of superficial characteristics that have no mathematical significance. (Carpenter & Moser, 1982)

There was, in fact, a brief time when the term "algorithmic learning" had more currency than it does today, at a time when information processing algorithms were enthusiastically regarded as models of cognitive processes. Today such models are competing with non-algorithmic connectionist models. For example, Suydam (1975) claimed that algorithmic learning involved more than just the learning of specific algorithms. It involved "learning-how-to-learn", generalizing from specific skills to broader process applications.

In summary, from an information-processing perspective, algorithmic performance cannot be regarded as purely associationist/behavioristic. The traditional controversy in mathematics education has been primarily the difference in emphasis placed on rote learning, skill acquisition, drill and practice, procedural knowledge, on the one hand, versus conceptual knowledge and understanding mathematical relationships, on the other hand. These are not categorical distinctions but matters of emphasis, since few educators today would deny the importance of both skill acquisition and understanding in mathematics.

2.2 Skill Acquisition

Schneider & Shiffrin (1977) found in their studies of controlled and automatic human information processing that consistent practice leads to automated processes where an input triggers a response sequence operating independently of the operator's control. This requires no attention or conscious processing as opposed to controlled responses that are not yet adequately practiced. The controlled responses require attention, use limited short term memory, and tend to be serial.

Anderson (1982, 1987) theorizes that a developing skill proceeds in two stages: a declarative stage in which facts about a skill domain are recalled and interpreted, and a procedural stage in which such declarative knowledge is embodied or compiled into procedures for performing the skill directly without having to recall and interpret facts. Declarative knowledge is encoded in a propositional network and procedures are encoded as "productions" (conditionaction statements). Within these encodings are two subprocesses: proceduralization, which embeds factual knowledge into productions, and composition, which collapses sequences of productions into single productions. Further learning processes --- generalization, discrimination, and production strengthening --- operate on a skill to make the productions more selective in their range of operations.

He believes that general problem solving skills (including what we have been referring to as "understanding" in mathematics) are forms of loosely organized declarative knowledge:

> The ACT* theory contains within it the outline of an answer to the epistemological question: How does structured cognition emerge? The answer is that we approach a new domain with general problem solving skills such as analogy, trial-and-error search, or means-ends analysis. Our declarative knowledge system has the capacity to store in relatively unanalyzed form our experiences in any domain, including instruction (if available), models of correct behavior, successes and failures of our attempts, and so on. A basic characteristic of the declarative system is that it does not require one to know how the knowledge will be used in order to store it. This means that we can easily get relevant knowledge into our system but that considerable effort may have to be expended when it comes time to convert this knowledge to behavior. (Anderson, 1987, p.206)
>
> Numerous experimental results may be predicted from this conception of skill organization and skill acquisition. These include predictions about transfer among skills, differential improvement on problem types, effects of working memory limitations, and applications to instruction. The theory implies that all varieties of skill acquisition, including those typically regarded as

> inductive, conform to this characterization. (Anderson, 1987, p.192)

Lesgold (1984) takes a similar tack towards what he calls acquiring expertise. During the learning of a complex procedure, pieces of the procedure become automated. If they execute in a fixed sequence, they can be composed into longer sequences, but if their sequence is not yet constrained, then thinking tends to be chaotic, somewhat like Selfridge's (1959) "pandemonium" model; the pieces of the procedure compete for attention and for placement in the sequence. He believes that "complex tasks involve multidirectional flows of control between procedurally and declaratively driven components" -- in other words, skill acquisition does not always flow one way from declarative to procedural knowledge but that proceduralization of some subprocesses leads to new declarative knowledge. Related to this is his suggestion that a verbal plan can help in the composition of isolated procedures into a linear sequence. Building the correct procedural sub-sequences guides "the development of systematic procedures from incompletely organized pandemonia of fragmentary productions".

Lesgold also believes that "representation construction" is needed for acquiring expertise, the ability to "see" relevant features in context, as would be required, for example, in expert interpretation of

X-ray plates in medical diagnosis. He makes another interesting distinction between the knowledge that comes from a variety of experiences (e.g. the chess master who seldom encounters identical game situations) and the knowledge that comes from repetition or practice (e.g. the long distance runner who traverses the same course again and again). In the context of mathematics education this corresponds to solving novel problems versus drill and practice of exercises. "The ability to build mental representations of problem situations is a central capability that involves both variation and repetition."

The views of Anderson and Lesgold may be characterized as a bottom-up perspective of knowledge acquisition, that is, knowledge is built up by an accumulation and integration of detailed, specific knowledge. Other researchers, particularly those who study problem solving in mathematics (Schoenfeld, 1985), believe that problem solving itself proceeds in a top-down fashion from general principles and concepts down to details. Schoenfeld acknowledges there must be fundamental resources available to the problem-solver, such as domain specific facts and procedures, algorithmic procedures that can be reconstructed, and other easily accessible competencies, but overriding these relatively low-level processes are the higher-level, top-down processes of conscious control, strategies and plans (heuristics), and belief systems. Strategies and techniques for progressing through non-standard problems involve use of imagery such as drawing figures, representing a problem in some kind of notation, reformulating the problem or working backwards, testing and verifying solutions. High-level control implies global decisions regarding the selection and implementation of resources and strategies. Another characteristic of expertise is the problem solver's belief system -- the attitude that a solution to a problem does exist and can be found with persistence. Such affective and metacognitive aspects of acquiring expertise are not addressed by purely cognitive approaches such as those of Anderson (1983).

2.3 Memory in Mathematics

It seems curious that debate in mathematics education has often polarized in terms of memory vs. understanding when it is clear that these are educationally mutually supporting (Byers & Erlwanger, 1985). Basic findings in memory research suggest their importance in learning mathematics. Much of the research on memory skills focuses on conscious strategies for encoding and retrieving information (Glass & Holyoak, 1986). There are many memory techniques: general techniques, such as rehearsal, use of imagery, and finding organizing principles; specific techniques, such as chunking, natural language mediation, semantic elaboration, and outlining; and quite special serial-order mnemonics, such as the method of loci and the pegword method . The method of loci involves associating the items to be remembered with an already remembered sequence of imaginary locations. The pegword method involves associating in vivid images the items to be remembered with an already remembered sequence of rhymes.

On the other hand, excessive reliance on such memory techniques is made at the expense of understanding the structures underlying the rules, formulas, and algorithms of mathematics. Mathematical structures and operations are not random assemblages, like word lists, to be recalled by some mnemonic technique. Mathematics does not require memorization in this sense, since instead of being remembered, many principles and relationships may be deduced and derived from other well-remembered relationships.

Madell (1985) describes informal invented methods for solving addition and subtraction problems in column arithmetic. He delays teaching the standard algorithms for a year while encouraging the development of invented methods of grouping and combining numbers. He wants the solution of problems to depend on the child's reasoning. One advantage is that there is a reduced need for a large store of memorized addition and subtraction facts. He sees another advantage in the freedom that teachers will have in spending more time on meaningful learning and less time on repetition.

> Eventually, of course, all the facts must be learned. But the early focus on memorization in the teaching of arithmetic thoroughly distorts in the children's minds the fact that mathematics is primarily reasoning. This is often difficult, if not impossible, to undo. (Madell, 1985)

Memory plays an essential role in understanding mathematics. Byers & Erlwanger (1985) in a review article on memory in mathematics understanding suggest that "a major source of mathematical errors should be sought in memory transformations and subjective organization". Important questions are what is remembered and how, by those who understand mathematics and by those who do not. Some indication of how material is well remembered has been known for some time: that it be organized and rendered "meaningful". For example, Bruner (1962) emphasized the role of organization in memory. Unless detail is encoded in memory as a structured pattern, it is rapidly forgotten.

> Organizing facts in terms of principles and ideas from which they may be inferred is the only known way of reducing the quick rate of loss of human memory ...... What learning general or fundamental principles does is to ensure that memory loss will not mean total loss, that what remains will permit

> us to reconstruct the details when needed. (Bruner, 1962)

Skemp (1987) sees consequences for remembering in the kind of mathematical understanding acquired by students. He distinguishes two types of mathematical understanding:

<u>Instrumental understanding</u>: recognizing a task as one to which a rule or formula may be applied. It is easier to understand than relational understanding. One can get the right answer more quickly. The rewards are more immediate and apparent. However, it is more difficult to remember all the specific rules and formulas and under what circumstances they are to be applied. For example, division by a fraction is understood instrumentally as "turn it upside-down and multiply".

<u>Relational understanding</u>: recognizing a task as one related to an appropriate schema. Although more difficult to learn, it is easier to remember. Rules and formulas are remembered as parts of a connected whole. It is adaptable to new tasks and motivates exploration into new areas of mathematics. For example, division by a fraction would be understood relationally as "the number of times the fraction is contained in the dividend".

Quite similar distinctions are made by Hiebert (1986) who contrasts procedural knowledge and conceptual knowledge. He and other researchers are generally in agreement that acquiring relational understanding or conceptual knowledge has a greater importance in learning mathematics than instrumental understanding or procedural knowledge. Most classroom practice, however, for a number of reasons, continues to revolve around acquiring procedural knowledge.

Byers & Erlwanger (1985) believe that this emphatic support by researchers for teaching relational mathematics has resulted in "unfortunate" attitudes towards the issue of memory in mathematics that have yet to be fully addressed. To what extent, conversely, does understanding depend on memory? Does the learning of principles invariably reduce the quantity and complexity of mathematical material held in memory? Can a "principle" be transformed in memory into a blind rule, thus resulting in a loss of mathematical understanding?

An aspect of memory that has direct bearing on mathematics learning is the way material is subjectively organized at the time of encoding and transformed at the time of retrieval or while stored in memory. This is implicated when errors, distortion, and misconceptions occur. Memory is not primarily detailed but schematic; even key details may not be encoded (Bartlett, 1932). New material is assimilated to a student's existing schemata:

> We suggest that many errors are due to attempts by students to simplify mathematical material. The student tries to introduce his own unity, coherence and consistency into material he has learned at different times, and to do so on the basis of hypotheses which appear to him to be both simple and sensible .... instances of Bartlett's "effort after meaning".
>
> Remembering mathematics is a more complex task than remembering a picture or story. [While this claim is moot, these tasks certainly are qualitatively different --- H.E.] For one thing, mathematical symbolism is replete with significant detail. For another, a mathematical statement, whether propositional or algorithmic, is already a precis. Although the meaning of such a statement has to be distinguished from its expression, small changes in wording [symbols] may turn a true statement into a false one, while small changes in procedure often result in wrong answers to problems. Few students are capable of paraphrasing a mathematical statement correctly, making the reproduction of definitions and the statement of theorems into difficult examination questions even at the university level. (Byers & Erlwanger, 1985, p.276)

In a comprehensive study of fourth grade children suffering from "mathematics difficulty" (MD) -- defined as children performing poorly in school math but normal in intelligence -- Russell & Ginsburg (1984) found MD children to be "essentially cognitively normal", similar to younger peers. Such children were not seriously deficient in key mathematical concepts and skills and were capable of "insightful" solutions of simple word problems. The "dramatic" exception was that MD children displayed severe difficulty in recalling common addition facts. Russell & Ginsburg consider this a surprising finding since rote acquisition of number facts would seem to be among the simplest of mathematical tasks. Small wonder that classroom teachers emphasize rote learning and drill and practice!

Information processing models of cognition generally assign two components to memory: long term memory (LTM), characterized by unlimited capacity and permanent storage, and short term memory (STM), characterized in small children by small capacity (3 to 5 "chunks"), ease of retrieval, and impermanent storage. Some researchers, for example Greeno (1973), have proposed another low capacity memory structure called "working memory", where data supplied by STM and LTM are organized for the task at hand. The limited capacity of working memory suggests that this component is readily overloaded: when the amount of the material being processed in working memory exceeds its capacity, some of the material is lost.

Case (1982) offers a similar hypothesis but with somewhat different terminology in place of LTM and STM. He refers to a "central coordinating or processing capacity", which becomes a key feature in his theory of cognitive development in the child. He believes the development of cognitive abilities is parallel across various domains of activity, including mathematical ability. A child's transition from one stage of learning to the next in any given domain depends not just on experience in that domain, but on the growth of some central coordinating or processing capacity. The ability to coordinate a certain number of elements at one stage is a prerequisite for assembling the operations at the next higher stage. Case suggests that what determines the rate of growth of processing capacity is the rate of increase of operational efficiency, given that total capacity is fixed. Operational efficiency is thus a function of both maturation and practice.

He believes the instructional implications are:

1. Match instruction to students' current developmental level.

2. Minimize the processing load during developmental transition.

3. Ensure that the child's basic operations are as efficient as possible by providing sufficient practice.

Memory capacity or deficit is not the only source of mathematics learning difficulty. Other sources may be deficiencies in logical reasoning, attention span, misconceptions ("bugs"), or lack of understanding.

2.4 Addition and Subtraction

Young children entering school at the age of 5 or 6 are known to bring with them informal knowledge of arithmetic (Ginsburg, 1980). They are able to solve simple addition and subtraction problems, often based on counting, even before they have been drilled in number facts or taught the standard algorithms.

> Young children often employ invented strategies; they do not always solve mathematics problems in the way the teacher intended. Instead, the child often devises a strategy which is partly of his own making. The invented strategy is usually a hybrid, a mixture of informal methods like finger counting and schooled procedures. The invented strategy reflects the child's contribution to the work of understanding. And often the child's input (for example regrouping) is so fundamentally sound that it can be used as the basis for formal instruction. The teacher can, in effect, build on what the child already knows. (Ginsburg, 1980)

Steffe (1983) contrasts the "mature" forms of school algorithms with immature child-generated forms, which may be regarded as comprising much of children's arithmetic knowledge, their "operative schemes" or mental structures.

Counting may be considered a prototypical algorithm. It is the first formal mathematics that a child usually learns before entering school. It displays features characteristic of all algorithms. It has been studied intensively (Gelman & Gallistel, 1978; Steffe, von Glasersfeld, Richards, & Cobb, 1983) as a means of discovering some of the principles underlying the child's developing cognitive abilities. According to Gelman and Gallistel, ability to count develops with the acquisition of implicit knowledge of counting principles -- the one-to-one principle, the stable numberword order principle, the cardinal principle, the abstraction principle, and the order irrelevance principle. The meaning of the counting algorithm may be said to reside in these principles. Here syntax and semantics are not separable, that is, each of the principles of counting may be regarded as embodying both procedural knowledge and conceptual knowledge.

Counting is basic to subsequent arithmetic knowledge. Much of school mathematics may be understood as an elaboration of counting. In particular, addition and subtraction may be seen as forward and reverse counting, and solving an arithmetic problem may be rendered as a question of what it is that needs to be counted. Stated this way, the transition from solving arithmetic problems informally by counting to solving multidigit addition/subtraction problems by standard algorithms would seem to be an easy one. However, for many children it is quite difficult and is often the beginning of a persistent pattern of failure and disaffection.

Addition methods of first- and second-grade children are still changing and unstable and, to a large extent, based on counting. Houlihan & Ginsburg (1981) described various addition methods in terms of counting vs. non-counting methods:

<u>Non-counting methods of addition</u>

Direct memory

Indirect memory (5 + 7 = 5 + 5 + 2 = 12)

Place value (32 + 45 = 30 + 40 + 2 + 5 = 77)

<u>Counting methods of addition</u>

Counting from 1 with concrete aids.

Counting from 1 without concrete aids.

Counting on from addend with aids.

Counting on from addend without aids.

Indirect: memory for combining, then counting.

Counting method not determinable.

Inappropriate method (guesses, alters).

Undeterminable answer.

Counting is the basis for subsequent understanding of multidigit numbers. Fuson (1989) reported that multidigit numbers may be represented by children in five different ways during their developing understanding of the operations of addition and subtraction. Such representations are often the source of many difficulties and misconceptions:

1. <u>Unitary representation</u>: This refers to the cardinal value of a multidigit number, the result of counting out a set of objects. The number is not yet understood as having a nested decimal structure.

2. <u>Named-value representation</u>: English number words (units, tens, hundreds, etc.) are used as labels for the digits but as yet have no quantitative meaning to the child.

3. <u>Multiunit sequence representation</u>: This refers to the solution of 2-digit addition problems by counting on by tens and units. For example, 35 + 47 could be solved by: "30, 40, 50, 60, 70, 75, 76, 77, 78, 79, 80, 81, 82".

4. <u>Concatenated single digit representation</u>: This treats a multidigit number as the sum of its individual digits, for example, 314 --> (3 + 1 + 4) --> 8

5. <u>Positional base-ten representation</u>: The position of the digits in a multidigit number conveys the place value of the digits.

Brown & Burton (1978) made an extensive study of the kinds of errors made in multidigit subtraction. They found that many errors (approximately 40%) could be explained as the result of "buggy" algorithms, that is, the application of an incorrect procedure in a consistent principled way. Resnick & Omanson (1987) cite two theories proposed to explain the origin of subtractive bugs: One by Young & O'Shea (1981) suggests that children either forget or never learned the standard school subtraction algorithm. The other by Brown & Van

Lehn (1980) proposes that children employ "repair" algorithms to repair incomplete or inappropriate procedures in order to overcome an impasse which resulted from forgetting or failure to learn. Repair algorithms are actions to try when the standard action is not known or forgotten. Resnick & Omanson (1987) note that these theories concern the surface structure of the procedure and not the principles underlying subtraction, particularly the place value system.

Similar findings of error frequencies in solving addition problems have been reported by McDonald, Beal, & Ayers (1987) who used computer software to diagnose 554 errors made by 51 subjects taking a 50-item test. Procedural errors accounted for 51% of the errors; 17% were errors of basic addition facts; and 32% were errors not identified by the software. Typically, there are several times as many procedural errors as there are number fact errors.

Understanding place value does not come easily to primary grade children (Kamii, 1986; Ginsburg, 1977). Kamii believes place value is difficult because children engage in a long process of constructing a system of tens on a system of ones. Initially, children understand numbers as a counted sequence; later they understand them as groupings of ten. This reflects the Piagetian view that understanding is a synthesis of ordering and hierarchical inclusion, an understanding of part/whole relationships (Inhelder & Piaget (1964). A further difficulty is that place value involves multiplication (e.g. sixty-one means six times ten and one more, which is not a simple extension of addition. When she asked children to count a heap of approximately 100 chips, she observed a progression from first-grade children who counted by ones and twos to second-graders who grouped the system of ones into heaps of ten and counted the ten-heaps and ones left over.

Such physical embodiments (manipulatives) of verbal or written numeric symbols have long been used to convey meaning in elementary mathematics, particularly the concept of place value (Dienes, 1963; Resnick & Omanson, 1987; Fuson, 1989; Fuson & Briars, 1989). Fuson & Briars (1989) found that 1st- and 2nd-graders demonstrated meaningful multidigit addition and place value concepts. The children could add large multidigit numbers when taught in the context of using both base-ten blocks to embody the English named-value system and digit cards to embody the positional base-ten system of numeration. Fuson & Briars employed a multi-representational board displaying base-ten blocks and their corresponding named-values and written numeric symbols. They emphasized that as each column on the board is added, recording in symbols should occur immediately after each move of objects so that the link between operations on objects and operations with symbols is clear.

Resnick & Omanson (1987) outlined place value principles underlying written subtraction, but which apply as well to written addition:

1. Additive composition of quantities. All quantities are compositions of other quantities (e.g. 7 is composed of 3 and 4, or 2 and 5, etc.)

2. Conventions of decimal place value notation. Each position in a multidigit number represents a higher power of ten. Each is limited to a value of 9 or less and thus constrain the compositions representing quantity. For example, the number 624 is composed of 6 hundreds, 2 tens, and 4 units.

3. Calculation through partitioning. This is the principle that permits written addition or subtraction to be done column by column. When multidigit numbers are added together, units are added to units, tens are added to tens, etc.

4. Recomposition and conservation of the partial sum. This principle leads to the "carry" procedure. For example, in adding 37 and 56, the sum of the units (7+6) is greater than 9, namely 13. The 13 is recomposed into 10 + 3; the 3 is the number of units in the sum; and the 10 is "carried", that is, added to the column of tens, thus conserving the total value of the partial sum, 13.

At what point in the mathematics curriculum does multicolumn addition first appear? In a comparison of curricula from Japan, mainland China, the Soviet Union, Taiwan, and the U.S., examining widely used textbooks, Fuson, Stigler, & Bartsch (1987) found that addition and subtraction of two multidigit numbers (2 digits ± 1 or 2 digits with trading from ones) started in the U.S. at grade 2.5 (about where this study begins) and addition and subtraction of 3 digits ± 2 or 3 digits with trading from tens started at grade 3. The other countries introduce trading earlier (up to a year earlier) and include in their texts activities supporting a specific method of solving problems with sums and minuends to 18. Solution of such problems is necessary for solving multidigit problems with trading.

2.5 Review of Methodology

Efforts to understand how children acquire their knowledge of arithmetic generally involve methods to establish their level of knowledge before and after some instructional treatment. The more commonly used measures are traditional written tests, observations of children manipulating concrete objects, and analysis of protocols, that is, transcripts or tape recordings of interviews (Ginsburg, Kossan, Schwartz, & Swanson, 1983). Interviews may be either structured, where probing questions are planned in advance, or take the form of an unstructured dialogue where the child's responses guide the questioning. Another newly developing method with these features used in this study is the interaction between a child, an instructor/researcher, and computer software.

Microcomputers proliferated in the schools in the 1980s and are playing an increasingly important role in some aspects of mathematics and language education (Reiser, 1987). Although a revolutionary role for computers in education has not yet materialized in the form as envisioned earlier (Papert, 1980), they have already demonstrated their usefulness and versatility in education as "tutor, tool, and tutee" (Taylor, 1980). For example, they appear as tutor in drill-and-practice programs and in intellectually challenging simulations; as tools for calculations and for word processing; and as "tutee", as a means of learning programming skills and instructing the computer in its own performance. They are being used to diagnose children's computational errors (Janke & Pilkey, 1985; McDonald, Beal, & Ayers, 1987). In recent years they are appearing as "intelligent tutors" fashioned around research in artificial intelligence (Sleeman & Brown, 1982).

In this study a microcomputer is used as both a "tutor" and a research tool, providing on-screen assistance and software guidance to the learner while capturing performance data for later analysis. The display of concrete representations of the "meaning" of the steps in the multicolumn addition algorithm offers the learner alternative opportunities for insight, different from those ordinarily offered by a textbook or workbook or by physical manipulatives. This is an example of a class of software that Dickson (1985) describes as designed to juxtapose two or more symbol systems. Users are encouraged to move back and forth between the systems, thereby promoting insight and understanding.

Most microcomputer applications in mathematics education involve two-way interactions between learner and computer. This study, however, is based on a three-way interaction between student, computer, and instructor. Here, the instructor (researcher), and not the software, provides the kind of support in the form of suggestions, prompts, and questionings, which intelligent tutors (that is, computers) some day may provide.

In this section on review of methodology, the work of Resnick & Omanson (1987) in particular is discussed in detail, since their studies most closely correspond to the design and intent of this study. They have attempted to establish the nature and extent of children's knowledge of the principles of subtraction in both written and non-written systems. They wondered whether children already knew a great deal about subtraction principles in the context of concrete, non-written systems, such as coinage or decimally coded (Dienes) blocks, but didn't know how to apply this knowledge to numeric symbols. To find out, they tracked the performance of ten third-grade children -- 5 boys and 5 girls -- over the course of the school year in tasks that tested their knowledge of subtraction with decimally coded blocks and with written numbers. Here is a brief outline description of the tasks employed to assess knowledge of subtraction with blocks:

A. Conventions of decimal coding

1. Name the value of individual blocks.
2. Read a display of concrete representations.
3. Construct a concrete display of a number.

B. Principle of recomposition (or regrouping)

1. Show a quantity in two ways.
2. Use a trade procedure (e.g. exchange ten one's for one ten) in subtraction with blocks.
3. Rebuild a block display with more of a "denomination" (e.g., show 34, consisting of 3 tens and 4 ones, with more than 4 ones.

And to assess knowledge of written numbers:

A. Conventions of decimal coding

1. Compare the value of the same digit appearing in two different columns.
2. Show the value of a digit using blocks.

B. Arithmetic procedures

1. Solve written addition problems with carrying.
2. Solve written subtraction problems with borrowing.

C. Principle of recomposition (or regrouping)

1. Name the value of the carry mark.
2. Name the value of the borrow mark.

Their findings were that the children had better command of value conventions in block representations than of those in written representations. Although they could use blocks to represent 2- and 3-digit total quantities, they could not use them reliably to represent individual digits. They showed good understanding of recomposition in blocks but were unable to assign appropriate values to written borrow and carry marks, suggesting that recomposition principles were not being applied in written arithmetic. They seemed to know that they could decompose <u>numerals</u> but didn't understand that they were actually decomposing <u>quantities</u>.

To discover what kinds of procedures children were using to solve single digit addition problems, Groen & Parkman (1972) measured response times. They proposed a model that assumed the existence of a mental counter with two operations, setting the counter and incrementing the counter. An addition problem presented in the form m + n may then be solved in several different ways:

1. The counter is set to zero. Both addends are counted (added) by increments of one.

2. The counter is set to m (the left number). The right number n is added by increments of one.

3. The reverse of (2).

4. The counter is set to which of m or n is the greater and the remaining number is added by increments of one.

5. The counter is set to which of m or n is the smaller and the remaining number is added by increments of one.

Mean response times were plotted as a function of the number to be incremented. Which procedure is most likely being used may then be inferred from the degree of correlation.

Model 5 above, when averaged over all subjects, was found to be the most likely strategy being used. On occasion, there were significantly low response times, for example for "ties" (where m = n), which may be accounted for by assuming the fast recall of number facts.

The Resnick & Omanson (1987) studies extended this response-time approach to infer whether recomposition and place value principles are being used by children in the addition of larger, 2-digit numbers. Assuming a mental counter procedure, four possibilities call on recomposition and place value principles to varying degrees when a 2-digit number and a 1-digit number are added (in the form of m + n, where m is a 2-digit number):

1. Minimum of the Addends. Reaction time would be a function of the single digit number. No understanding of the decimal system of numbers is required.

2. Sum of the Units. Reaction time is a function of the sum of the two units digits. The counter is set at the beginning of the decade of the 2-digit number (e.g. 23 + 8 is recomposed into 20 + 3 + 8). This procedure reflects an understanding of the composition of 2-digit numbers but not full appreciation of recomposition.

3. Minimum of the Units. Reaction time is a function of the smaller of the two digits in the units column (e.g. 23 + 8 is rearranged into 28 + 3). This procedure indicates that the child may understand how numbers may be recomposed.

4. Mental Carry. This procedure mimics the carrying procedure for written arithmetic. It is difficult to discriminate this procedure from the Minimum of Units (No.3 above) on the basis of reaction times because the units are added together initially. However, if reaction times are significantly lower when the units are doubles that are being added (e.g. 28 + 8) then this would suggest that the child was using Mental Carry.

They concluded that relatively few primary children use procedures that apply recomposition principles to the decimal structure of the counting numbers.

> Many children's difficulty with place value in written arithmetic may result not from a total absence of knowledge of the relevant principles, but from an inadequate linking of the principles with the symbols and syntax of the written algorithm. (Resnick & Omanson, 1987, p.71)

This suggested that they develop instruction that links principles with instruction. Accordingly, they tested a method of instruction called <u>mapping instruction</u>, requiring a child to do subtraction problems both with Dienes blocks and in written symbols, maintaining a step-by-step correspondence between the blocks and the written symbols. Resnick (1982) had earlier identified three levels of mapping:

1. Code mapping: Shape or color of the concrete materials codes the same information as position (column) in the written numerals.

2. Result mapping: Procedures in the concrete materials yield the same answers as procedures in the written system.

3. Operations mapping: Operations in the concrete system are identified as equivalent operations in the written system.

Mapping is thus one explanation to account for the results, which were encouraging. Understanding developed with blocks in a concrete way is transferred to written arithmetic. "Semantic knowledge initially embedded in the blocks algorithm is applied to the rules for writing so that the newly enriched knowledge structure then eliminates bugs" (Resnick & Omanson, 1987), and justifies and explains the steps in the algorithm.

An alternative explanation of how mapping instruction works is that it enables the routine to be rehearsed without making errors (the result of "<u>prohibition instruction</u>"). "The pairing of each step in the blocks with its parallel step in the algorithm may prohibit wrong operations in the writing .... and provides high feedback to override an entrenched bug." (Resnick, 1982).

To explore these alternatives, they compared mapping instruction with an explicit form of prohibition instruction, which consisted only of practicing the written algorithm with no incorrect steps permitted and no Dienes blocks used. They found, with some disappointment, that neither mapping nor prohibition instruction was very successful in correcting bugs.

Could the failure of mapping instruction to correct bugs be due to the incompleteness of children's understanding after instruction? Resnick and Omanson then looked at the relationship between an individual's level of understanding and his/her performance on the written subtraction procedure. They identified five levels of understanding based on understanding of place value and composition principles to explain borrowing. Only a small minority of children who reached Level 5 (full understanding of place value and composition principles) were able on a delayed posttest to perform the subtraction algorithm without bugs. They concluded that "mapping instruction, in the form presented, is not effective in curing subtraction bugs, even when it induces understanding of the principles underlying the subtraction procedure."

To try to account for the great variability in learning the principles of subtraction in the course of mapping instruction, Resnick & Omanson investigated what factors are likely to determine who will learn. They found:

1. Differences in entering knowledge.

2. Differences in amount of instruction.

3. Differences in time spent on manipulating blocks.

4. Differences in time needed to master the steps of the mapping instruction plan.

5. Differences in the child's verbalization of quantities during instruction.

What seemed to characterize the learners from the non-learners was having longer interviews and using this added time to make more correct verbalizations of the quantities involved in borrowing. Resnick & Omanson rejected the notion that understanding is transferred directly from blocks to the written arithmetic system as a result of mapping instruction; it seems to be attention to the <u>quantities</u> that are being manipulated in both blocks and written symbols that produces learning. The children did not always call upon all of their relevant knowledge when calculating.

Furthermore, Resnick and Omanson believe that their mapping instruction did not fully address the issue between automated and deliberately controlled skills:

> If, when they are doing routine calculation, children do not represent the problem as involving

> quantities but only as digits to be manipulated, then there is no simple way for them to apply their newly learned principles. They must first interrupt their normal performance to re-represent the problem for themselves as one involving operations on quantities. But this means giving up all the efficiency of an automated skill and requires paying attention to every step. (Resnick & Omanson, 1987, p.92)

They suggest two general principles for mathematics instruction drawn from their current studies:

1. Early focus on the principles of a procedural domain might prevent buggy rules from becoming automated.

2. Instruction should be designed that invokes and maintains a reflective attitude towards how principles apply to each step of a calculation procedure.

Wearne & Hiebert (1988) draw a distinction between their own instructional approach which they call "semantic analysis" and mapping instruction. Semantic analysis begins with meanings of individual symbols, spending a major part of instruction on connecting symbols with referents; actions on referents are then used to generate procedures with symbols, even invented procedures. Mapping instruction develops a rationale for a standard algorithm by comparing step by step actions on blocks with the movement of symbols on paper; alternative non-standard but appropriate algorithms are less likely to emerge. But both approaches are commendable in trying to help students make sense of algorithms by connecting the rules with referents.

2.6 Summary

This literature review opened with a brief historical introduction to the decades long controversy between mathematics educators, those who emphasized behaviorist methods and those who emphasized understanding of mathematical relationships in instruction. Cognitive approaches to instruction have mitigated this controversy to some degree, but the controversy continues today in more sophisticated guise between those who believe mathematics learning to be a matter of skill acquisition and those who believe it to be the construction of meaningful mathematical relationships by the learner. However, these are not mutually exclusive positions.

Areas of literature relevant to the learning of algorithms were then reviewed, namely, skill acquisition, memory in mathematics, and educational studies in addition and subtraction. In skill acquisition, we focused primarily on the cognitive theories of Anderson (1983) and Lesgold (1984), which may be characterized as bottom-up perspectives. The role of memory in understanding and learning mathematics was explored, primarily in the work of Byers & Erlwanger (1985).

Material is remembered better when presented in some meaningfully organized way or is organized meaningfully by the learner. Case (1982) claims that total working memory capacity is fixed but that processing capacity increases with increasing operational efficiency, that is, when procedures being learned become automated.

Finally, studies in addition and subtraction and some research methodologies were reviewed, particularly the work of Resnick & Omanson (1987), who are concerned with the inability of children to link principles with the symbols and syntax of written algorithms.

CHAPTER 3

METHOD

In this study computer software has been designed as an instructional aid in the teaching of the multidigit addition algorithm. This is not computer aided instruction (CAI) in the usual sense in which the student interacts solely with the computer. Here, the student with light-pen in hand, the computer with its screen displays, and the teacher (researcher) who instructs and prompts, are involved in a 3-way interaction.

The research focused on differences between three versions of the software. The control group used the version which does not provide on-screen assistance for number facts; that is, the student must recall number facts while learning the algorithm. The assisted group used the version which does provide on-screen assistance for number facts; and the simulation group used the version which provides additionally, in the form of simulated blocks, an on-screen representation of the quantitative meaning of the symbol manipulations. Section 3.3.4 describes the software in detail.

Also described in this chapter are the population from which a sampling of subjects has been drawn, the sampling method, sample size, the relevant variables, the tests and interview questions that were used to characterize subjects before and after treatment, and the research design.

A pilot study was conducted of 6 subjects drawn from a second grade class at the Marks Meadow Elementary School, Amherst, MA in order to debug and fine-tune the instrumentation and to standardize instructions.

3.1 Sample

The sample consisted of 44 second grade children drawn from two classes in the Morgan School, an inner city elementary school in Holyoke, Massachusetts. When the study began in January 1989, the children had not yet received formal classroom instruction in 2-digit column addition. Classroom instruction was based on a workbook entitled "Addison-Wesley Mathematics" (Eicholz et al., 1985). Pretests were individually administered consisting of questions about basic first grade arithmetic, understanding of place value, and written 2-digit column addition.

After the pretest, 5 children were dropped from the sample, either for doing very well and so not needing instruction in the topic or for doing quite poorly (particularly in first grade arithmetic) and so insufficiently prepared to begin 2-digit column arithmetic.

The remaining sample of 39 children were then divided into three statistically comparable groups, equalized for pretest scores, classroom membership (teacher), and sex. (During the course of the study, 3 more children were dropped -- two moved to another school and one was absent with an extended illness. The final sample consisted of 36 children in three groups of 12 each.) The groups were then randomly assigned as control group and two treatment groups. See Appendix A for the equalizing method used and statistical comparisons between the groups formed. Several methods of weighting the pretest scores were compared. No statistically significant differences were found among the groups regardless of the weighting method used, indicating that the groups were satisfactorily equalized.

The final composition of the three groups is shown in Table 3.1. The 36 subjects consisted of 20 girls and 16 boys from two 2nd-grade classrooms, 21 from Room G and 15 from Room F.

TABLE 3.1 Composition of the Groups

	Control	Assisted	Simulation	Total
Room G / Female	4	4	4	12
Room G / Male	2	4	3	9
Room F / Female	2	3	3	8
Room F / Male	4	1	2	7
Totals	12	12	12	36

3.2 Research Design

The research design was a simple controlled experiment with pretest and posttest to determine the effect of two treatments on learning the multicolumn addition algorithm. The independent variable is the group assignment: the control group and the two treatment groups. Dependent variables are the pretest and posttest scores.

In a conventional classroom setting children are instructed in 2-digit column addition on a chalkboard and are required to work examples on paper with pencil. In this experimental situation, examples are presented on a computer screen, and, after a brief demonstration by the instructor, the child works examples by touching a light pen to the digits displayed on the screen.
The control and treatment groups are all instructed in the algorithm in the same way in this medium. They differ only with respect to the kind of screen display, as follows:

Control Group: No on-screen assistance for number facts. The subject learns the algorithm and adds single digits mentally (or by counting). See Figure 3.1.

Assisted Group: On-screen assistance for number facts. As each subject in the assisted group touches each digit to be added, the cumulative sum appears on the screen in a "Memory Box". This is the on-screen assistance that allows the subject to learn the perceptual/motor aspects of the algorithm without having to recall number facts initially. The assisted group is the basis for testing the hypothesis that, with automation of the procedural aspects of the algorithm and therefore less cognitive demand on short term memory, effective learning is likely to occur. See Figure 3.2 for the screen display.

Simulation Group: Number facts assist with mapping display. This treatment is similar to that of the "assisted" group in providing on-screen number-fact assistance, but it also has an additional feature intended to provide the subject with the possibility of an insight into the meaning of the symbol manipulations. As the problem is presented, the value of each of the numbers to be added together is decomposed into tens and units and is represented as, or "mapped" into, an array of simulated blocks. Then, as the subject places each digit of the sum into its proper position, images of blocks appear in positions on the screen corresponding to units, tens, and hundreds. The purpose of this display is to demonstrate to the subject the connection between counting, which the subject presumably understands, and the addition algorithm. This is a variant of mapping instruction described earlier. Its purpose is also to test the second hypothesis -- that understanding the concepts supporting the algorithm leads to effective learning. See Figure 3.3 for the screen display.

All three groups work through the same 24 problems with instruction, commentary, and questions by the instructor. At the end of this instructional phase that occurs in three working sessions over 10 days, each subject is given a test to work 6 problems in the control group mode without on-screen number fact assistance and without any intervention or commentary by the instructor.

3.3 Instrumentation

This section contains the details of the various instruments used in the study: descriptions of the pre- and posttests; the problem set used in the instruction; hardware and software; and the instructional script.

3.3.1 Pretests and Posttests

The pretest and the posttest are exactly the same in content, but at least four weeks separates the administration of each of them. The tasks are similar to those employed by Russell & Ginsburg (1984) in their study of "mathematics difficulty" children and are scored one point for each fully correct response. A few additional problems after the posttest are given to each subject to get an indication of understanding of the algorithm and of any transfer to more difficult problems.

Introductory remarks

See the instructional script in Section 3.3.5 below.

Ability to count

1. "Start counting up from 14." (up to 25)
2. "Start counting up from 87." (up to 105)
3. "Can you count by 2s?"
4. "Can you count by 5s?"
5. "Can you count by 10s?"
6. "Can you count by 100s?"

Knowledge of number facts

Oral 1-digit addition

7. "How much is 4 and 2 ?" --- "How did you get that?"
8. "How much is 3 and 5 ?" --- "How did you get that?"
9. "How much is 11 and 6 ?" --- "How did you get that?"
10. "How much is 7 and 8 ?" --- "How did you get that?"
11. "How much is 13 and 0 ?" --- "How did you get that?"
12. "How much is 4 and 6 ?" --- "How did you get that?"
13. "How much is 6 and 4 ?" --- "How did you get that?"
14. "How much is 10 and 7 ?" --- "How did you get that?"
15. "How much is 5 and 50 ?" --- "How did you get that?"

Written (symbolic) addition

16. "Can you write down 6 plus 3 ?" -- "and the answer?"

17. "Can you write down 2 plus 9 ?" -- "and the answer?"

"Can you write down the answer to this?"

18. (Hold up card) 4 + 5 = ?

19. (Hold up card) 6 + 3 = ?

20. (Hold up card) 3 + 8 = ?

21. (Hold up card) 7 + 2 = ?

Read 2-, 3-digit numbers

"Can you read this?"

22. (Hold up card) 54 ?

23. (Hold up card) 776 ?

24. (Hold up card) 308 ?

Counting money

25. "How many cents are there in a dime?"

"How much money --- how many cents --- do we have here?"

(Spread coins randomly on table):

26. 3 dimes and 7 pennies

27. 2 dimes and 15 pennies

28. "Can you pick out 43 cents from all this money?"

(from a scattered array of dimes and pennies)

Place value

What does this digit mean?

(Hold up a card bearing a number and point to the digit)

1. "What does this (4) mean, or stand for?" 54
2. "What does this (5) mean, or stand for?" 54
3. "What does this (7) mean, or stand for?" 776
4. "What does this (7) mean, or stand for?" 776
5. "What does this (6) mean, or stand for?" 776
6. "What does this (8) mean, or stand for?" 308
7. "What does this (3) mean, or stand for?" 308
8. "What does this (0) mean, or stand for?" 308

Which number of a pair is larger? (Hold up card)

9. "Which number is larger or greater?" 522 288
10. "Which number is larger or greater?" 799 877

How many tens / hundreds?

11. "How many tens are there in 146 ?"
12. "How many tens are there in 52 ?"
13. "How many hundreds are there in 378 ?"
14. "How many hundreds are there in 529 ?"

Name tens

15. "What are four tens called ?"
16. "What are ten tens called ?"

Positional value of digit

17. "Here are two numbers. (Hold up card with: 32 73)
    Can you tell me the difference between the 3 here and the 3 here? Are these different kinds of 3 ?"

Decomposition

18. "What three numbers make up, add up easily to 658 ?"

(A deliberate hint is offered in the form of pauses and emphasis of voice: "Six hundred" [pause] "fifty" [pause] "eight")

Composition

19. "Can you add these numbers in your head ?"

(4 + 70 + 200)

Number proximity (1-digit)

20. "Here are two numbers on this card (2 7). And here are two numbers on this card (4 5). Which card has the numbers closer to each other ?"

Number proximity (3-digit)

21. "Here are two numbers on this card (436 448). And here are two numbers on this card (546 548). Which card has the numbers closer to each other ?"

Multicolumn Addition Problems

Each subject is given worksheets bearing 8 addition problems in horizontal format to solve. The problems are presented in horizontal format to ascertain whether the subjects is able to rewrite the problems vertically for easier solution. As follows:

1) 45 + 3 =

2) 13 + 46 =

3) 88 + 37 =

4) 96 + 7 =

5) 5 + 68 =

6) 26 + 38 =

7) 54 + 62 =

8) 84 + 67 =

Since 4 to 6 weeks elapsed between the pretest and the beginning of instruction on the computer, a "monitor check" was administered to each subject just before instruction to ascertain whether the addition algorithm had been learned in the interim. This check consisted of the following three worksheet problems: 86+42 57+18 56+78

3.3.2 Transfer and Correction Tasks

When the posttest was completed, each subject was asked to solve six additional problems which extended the algorithm to three 2-digit addends and to 3- and 4-digit numbers. Success in this task would indicate that near transfer is occurring. The "transfer" problems were presented in vertical format as follows:

(1)	(2)	(3)	(4)	(5)	(6)
68 42 +57	79 37 +16	407 +847	977 +221	2847 +3625	5474 +4378

Another task (called "correction" problems) involved asking each subject to "make believe that you are the teacher and I have just done these problems (handing the subject a sheet bearing 4 finished problems). Would you look them over and correct them if you find anything wrong and tell me why." The "correction" problems were presented as follows:

(1)	(2)	(3)	(4)
54	26	46	29
+38	+18	+37	+ 1
93	314	73	20

After these tasks, each subject is asked what the "carry" means, while pointing to a carry mark on one of the problems. Finally, each subject is asked what he/she liked or did not like about learning to add on the computer.

3.3.3 The Computer Problem Set

The following lists the types of problems that may be encountered in 1- and 2-digit addition and the specific 30-problem set administered to each subject:

<u>Types</u>

Type 1: One-digit addends, no carry.

Type 2: One-digit addends with carry.

Type 3: Two-digit plus one-digit addends, no carry.

Type 4: Two-digit plus one-digit addends with carry.

Type 5: Two-digit plus One-digit addends, sum > 99.

Type 6: Two-digit addends, no carry.

Type 7: Two-digit addends, carry from units column only.

Type 8: Two-digit addends, carry from tens column only.

Type 9: Two-digit addends, carries from both columns.

<u>Problem Set</u>

1)	2 + 1	Type 1
2)	3 + 2	Type 1
3)	5 + 4	Type 1
4)	40 + 24	Type 6
5)	15 + 72	Type 6
6)	82 + 16	Type 6
7)	33 + 5	Type 3
8)	48 + 14	Type 7
9)	73 + 52	Type 8
10)	25 + 7	Type 4
11)	78 + 79	Type 9
12)	15 + 72	Type 6
13)	93 + 9	Type 5
14)	19 + 25	Type 7
15)	16 + 4	Type 4
16)	41 + 84	Type 8
17)	96 + 7	Type 5
18)	63 + 47	Type 9
19)	69 + 2	Type 4

20)	88 + 55	Type 9
21)	92 + 8	Type 5
22)	61 + 92	Type 8
23)	54 + 9	Type 4
24)	95 + 7	Type 5
25)	47 + 89	Type 9
26)	24 + 57	Type 7
27)	66 + 62	Type 8
28)	56 + 38	Type 7
29)	35 + 73	Type 8
30)	93 + 38	Type 9

3.3.4 Hardware and Software

The hardware consisted of an Apple IIe computer into which was installed a Gibson (Koala) light pen. The light pen system includes both interfacing hardware which plugs into Slot #7 of the computer and software which permits commands for the light pen system to be embedded in a BASIC program.

The following describes the software written for this study as it applies to the control group. All input/output interaction by subjects with the monitor screen of an Apple IIe computer is done by means of the light pen.

When the pen is in "tracking" mode and its tip is held briefly (approximately 0.5 sec) against an illuminated portion of the screen, the coordinates of the pen's position are calculated and can be stored for later retrieval. Conversely, a display may be made to appear on the screen if the pen is held briefly at a position previously specified by the software. Input/output occurs only when the pen is held stationary for a very brief period of time (0.5 sec), indicating that the user has made a decision to point at a particular location on the screen. Such placement of the pen is called a "pen hit". In this way, all pointing responses can be captured and stored on a floppy disk for later analysis. The time between pen hits is also captured with a precision of ± 0.1 sec

There is a parallel between using the light pen on the screen and using a pencil on paper. Both the light pen and a pencil are used during the process of calculation primarily as pointing tools, "counting off" or tagging the numerals as they are processed. The light pen, however, unlike a pencil, does not "write", but with appropriate moves "picks up" and "lays down" the appropriate numeral.

The following is a typical sequence of activities during the subject-computer-instructor interaction:

See Figures 3.1 to 3.6 at the end of this chapter for screen layouts for the control and treatment groups.

At the top edge of a blank screen, one of the nine types of addition problems used in this study is displayed in horizontal format (e.g., 38 + 27 = ?), beginning with the simpler addition of One-digit numbers. There ensues a brief pause in order to allow the subject to read the problem and to express understanding of the nature of the task.

The problem is then presented in vertical format:

$$\begin{array}{r} 38 \\ +\ 27 \\ \hline \end{array}$$

The instructor explains that "we arrange the numbers this way so that we can add them together easily. We can add very large numbers this way easily, too."

Then an array of the 10 digits (0 through 9) divided into two rows appears in reverse video:

0 1 2 3 4

5 6 7 8 9

(The reason for putting the digits into two rows is space limitations on the screen.) These are the digits the subject will tap (with the light pen) and then place in the appropriate positions. The instructor explains: "These are the answer numbers you will pick to put into the answer place."

The subject points with the light pen to each of the digits to be processed in sequence, moving down the column of digits. A sound signal (beep) indicates whether the correct digit is being touched in the correct sequence (that is, according to the standard school addition algorithm). This is intended to mean that the digit is to be added mentally. Results of calculations on the column of digits are inscribed in the appropriate position by "dragging" the appropriate digit from the 10-digit array.

All of the light pen moves made by the subject and the timing of moves are captured in a software array. At the end of each worked problem, the contents of the array are transferred as a text file to a floppy disk for later analysis. Each subject is instructed in multicolumn addition through this medium in a series of 30 to 45 minute sessions on different days. The instruction ends when 24 problems have been processed. The remaining 6 problems of the 30-problem set are reserved for testing the subject's acquired skill in performing the computer algorithm.

3.3.5 Instructional Script

All three experimental groups were presented with the same 30-problem set drawn from a computer file (in order to standardize the type, number, and sequencing of the problems).

The following is the instruction given to each subject. Somewhat more elaborate instruction is given to the simulation concerning the blocks display.

Pretest: Session 1

A brief informal introduction: "Have you ever played with a computer? -- I hope this will be fun -- I've tried to make this computer help children to learn to add numbers -- How do you like doing number work and math? -- Before we get started, I'd like to ask you some questions about numbers and math so that I can find out where you might need some help in learning to add big numbers -- I think you can count up to a high number. How high do you think you can count?" -- (The pretest begins here.)

(After the pretest, the first interaction with the computer and the software begins. All subjects do the first 6 problems in the problem set. No carries are involved) "We're going to start now on the computer. You can type in your name. I'll do the first one to show you how to do it and then you can try it yourself. (The problem appears in horizontal format. The instructor reads off the problem.) Five plus two equals -- that's a question mark. The first thing we do is to rearrange the numbers, put the numbers up and down in a straight line.

(The computer does this. Then the number array (0-9) appears.) We'll use these answer numbers to put our answers in the answer place. (where the sum is placed)"

(Instructor describes light pen) "This is a special pen. It has a little hole in the end of it. At the bottom of the hole there's an electric eye that sees where you put the pen on the screen. You hold the pen straight out from the screen (demonstrates) and touch the screen. Now listen as I add the five and the two (Instructor demonstrates. A beep is sounded as each number is tapped in the correct sequence.) How much is five and two? -- Seven -- So I bring the seven up from the answer numbers (touches the 7 in the array), and I hold it in the answer place. (The seven appears in the answer place. A brief ascending tone scale is sounded to indicate a successful completion of the problem.) Now you can try it". (The first three problems are One digit additions. When a new problem is displayed in horizontal format, the subject is asked, "How do you read that?")

(The next three problems are double-digit additions with no carries.) (Demonstrates) "The way we do this one (namely, 23 + 41 = ?) is we first add the ones parts together and then the tens parts together. Notice the twenty three is two tens, or twenty, and three -- twenty three. Forty one is four tens or forty -- forty and one is forty one. When you add the ones parts, you put the answer here, and then we add the tens parts, the two tens and the four tens and put the answer here. (Demonstrates) -- Three ones and one is? -- four ones (Demonstrates) -- Two tens and four tens is how many tens? -- six tens -- So now how much is 23 plus 41? -- (Subject reads answer, 64) -- What does that six stand for? (pointing to the 6 in 64) -- six tens -- And what does the four stand for? -- four ones -- What is 6 tens called? -- sixty -- sixty and four make sixty four. That's how we add big numbers together." (The instructor reviews the procedure.)

(With both the assisted and simulation group, the instructor describes the "memory box". Instruction is identical to that of the control group, except that a "memory box" appears on the screen after the vertical layout of the problem appears. Then the following explanation is made:) "This is a memory box. This is where the computer will help you remember your addition facts as you go through the steps of the addition. Later on, after you have learned all the steps, you can try to do the addition without the memory box."

Instruction - Sessions 2 and 3

(The following is a typical script used with all groups when explaining the carry. The problem being solved is 88+55) (For the simulation group only: Point to the simulated blocks as they are being displayed on the screen) "This shows what 88 looks like. It is made up of 8 tens, or eighty, and 8 ones. Now what does 55 look like? -- 5 tens, or fifty, and 5 ones. Now you can start adding the ones parts -- 8 ones and 5 ones are? -- 13 ones --so you put the ones part of the 13 into the ones part of the answer place. Then you carry the one ten left over from the 13 up here into the tens place (pointing). The computer is doing the same thing (10 blocks are moved into the tens column.) Now you add up all the tens parts -- One ten and 8 tens and 5 tens are? -- 14 tens. So you put the 4 tens into the tens answer place and the computer does the same thing (in simulated blocks). Notice that leaves 10 tens left over from the 14 tens. Ten tens are one hundred, so we put or carry a one that stands for one one-hundred up here in the hundreds place. (Demonstrates) Now you add up all the hundreds parts. One one-hundred and blank is? -- one one-hundred. So you pick up a one and put it into the hundreds answer place. The computer does the same thing with the little blocks that you are doing with numbers. What is the answer? -- 143. What does 143 look like? (Pointing to the blocks in the answer place) It is made up of 100 blocks plus 40 blocks plus 3 blocks -- 143."

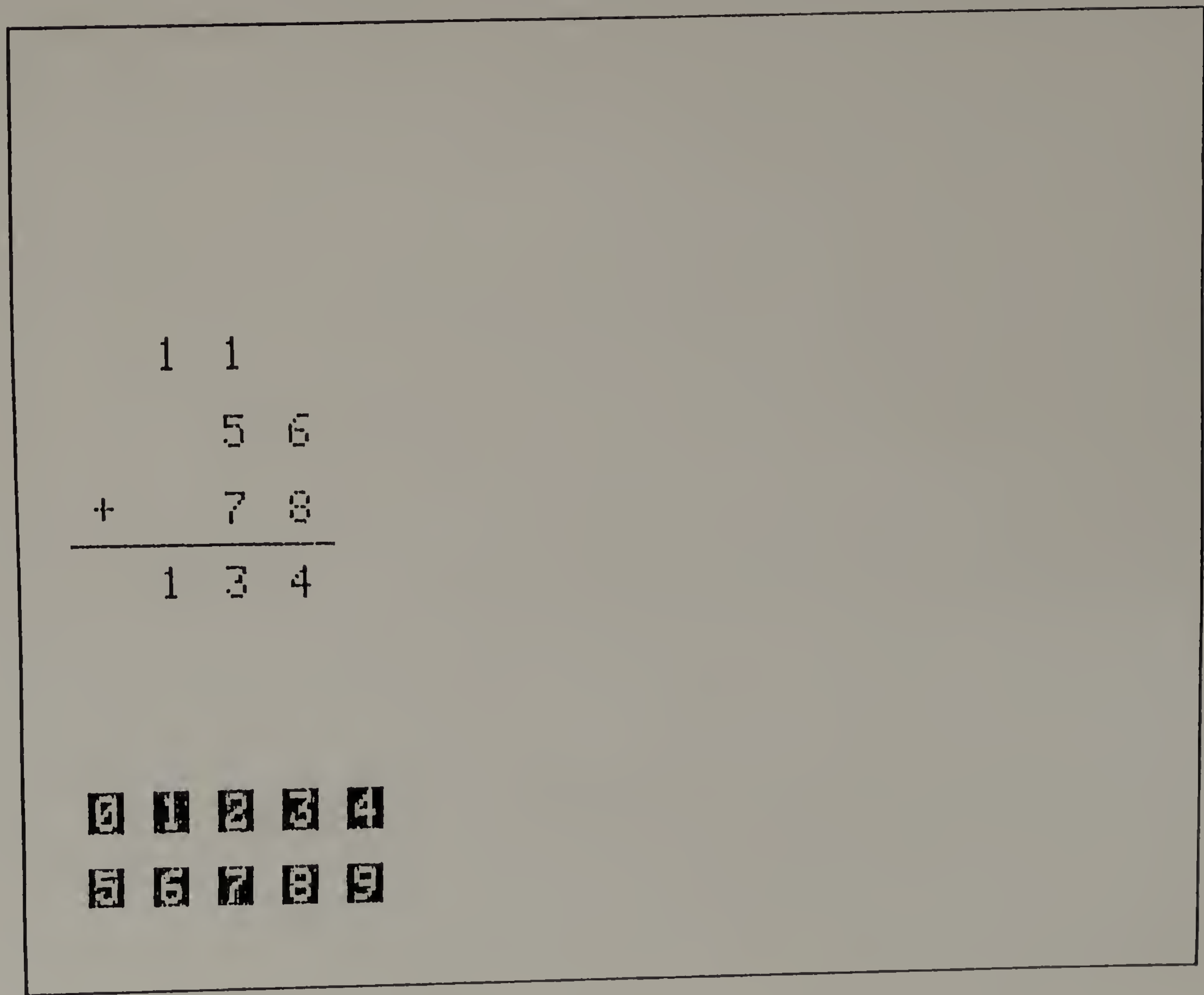

Figure 3.1 Screen display: control group

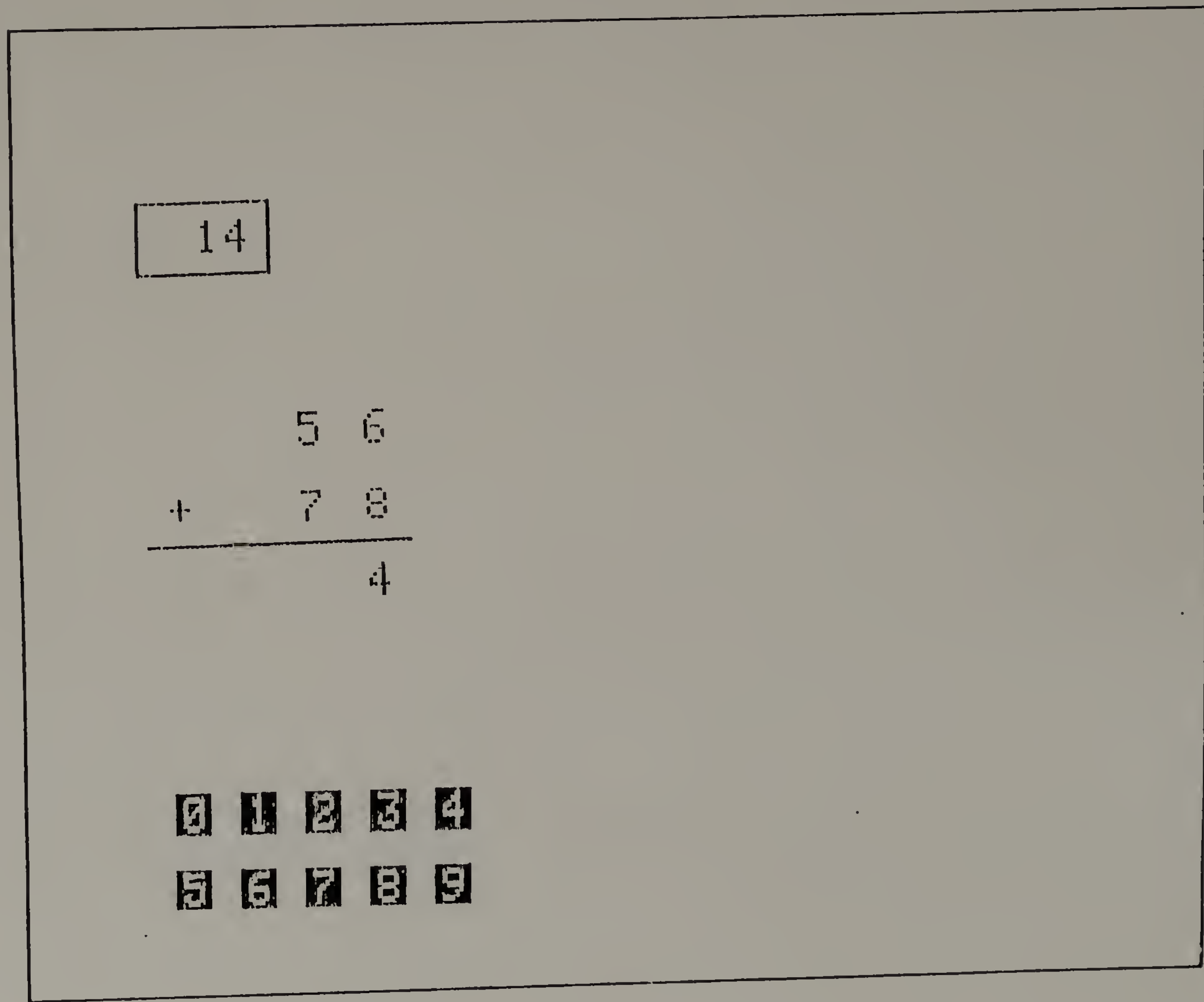

Figure 3.2 Screen display: assisted group

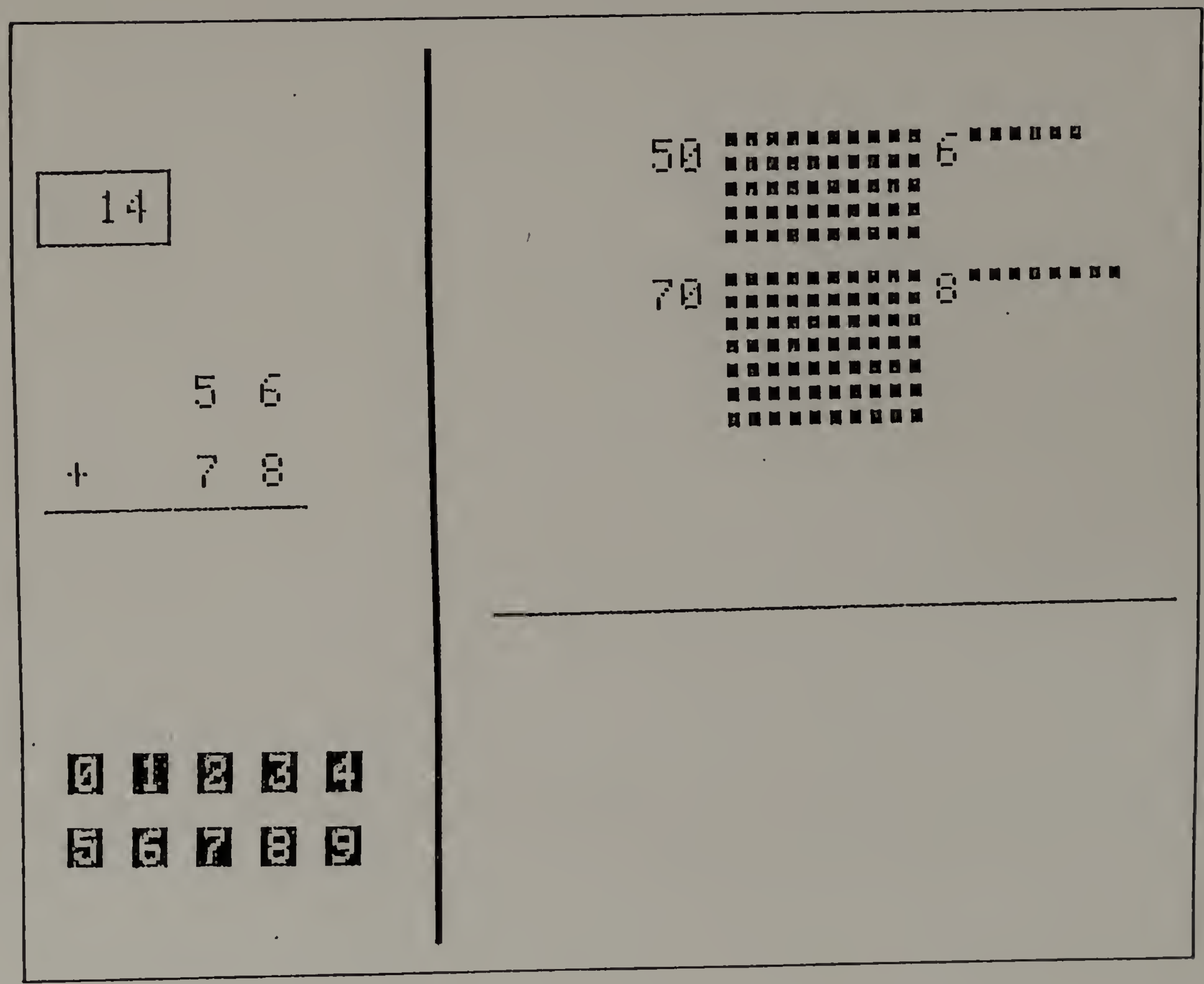

Figure 3.3 Screen display: simulation group

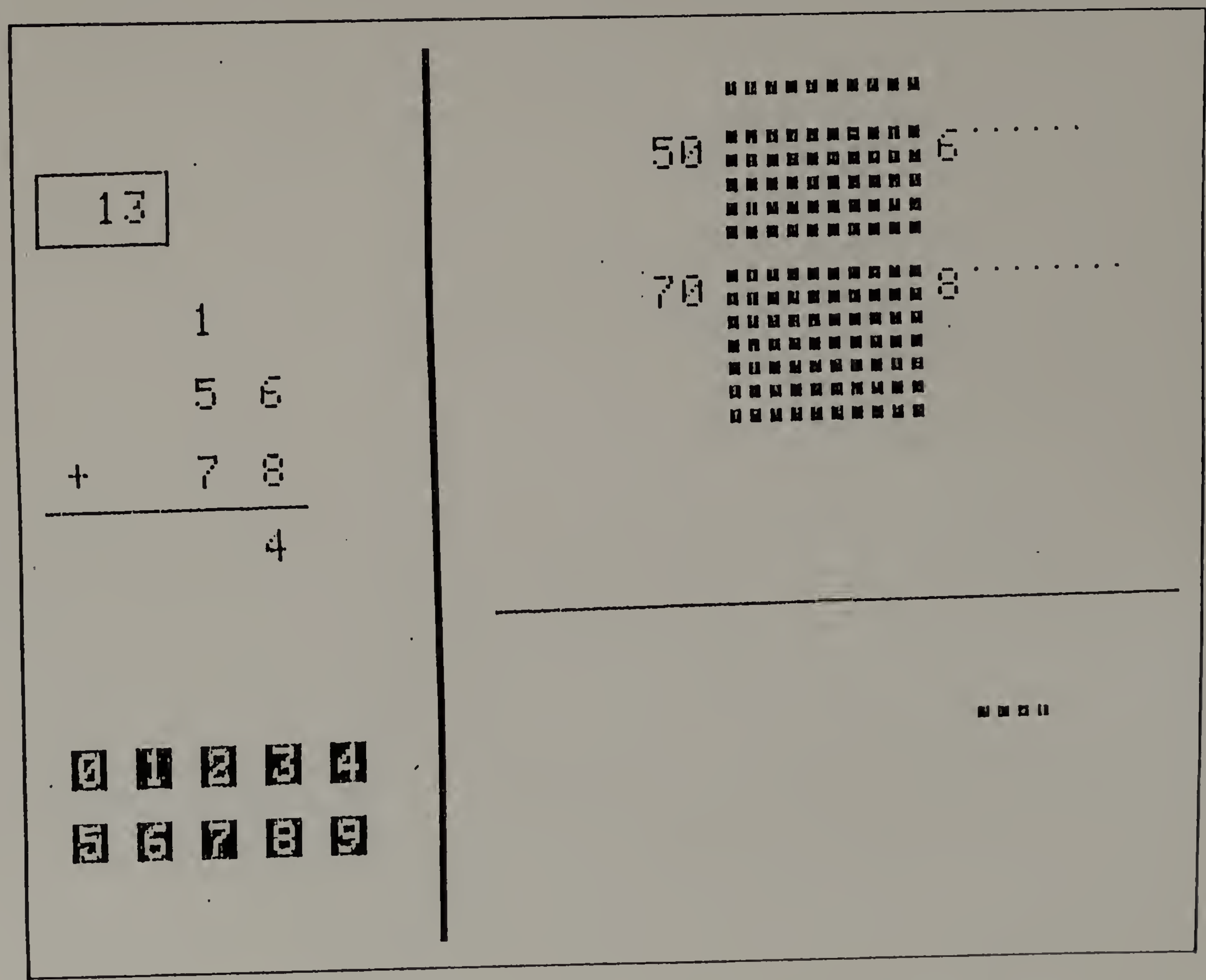

Figure 3.4 Screen display: carrying the ten

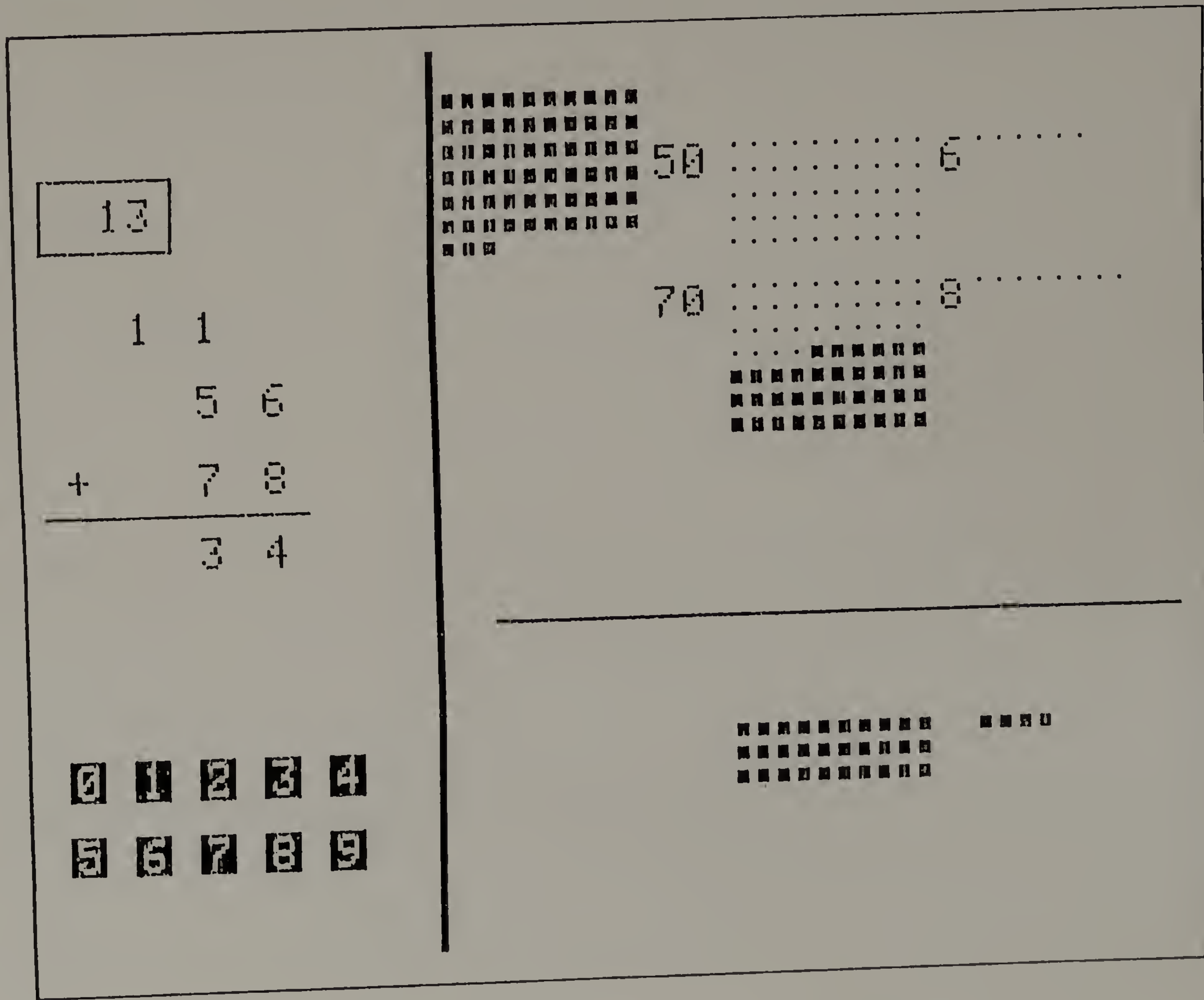

Figure 3.5 Screen display: carrying the hundred

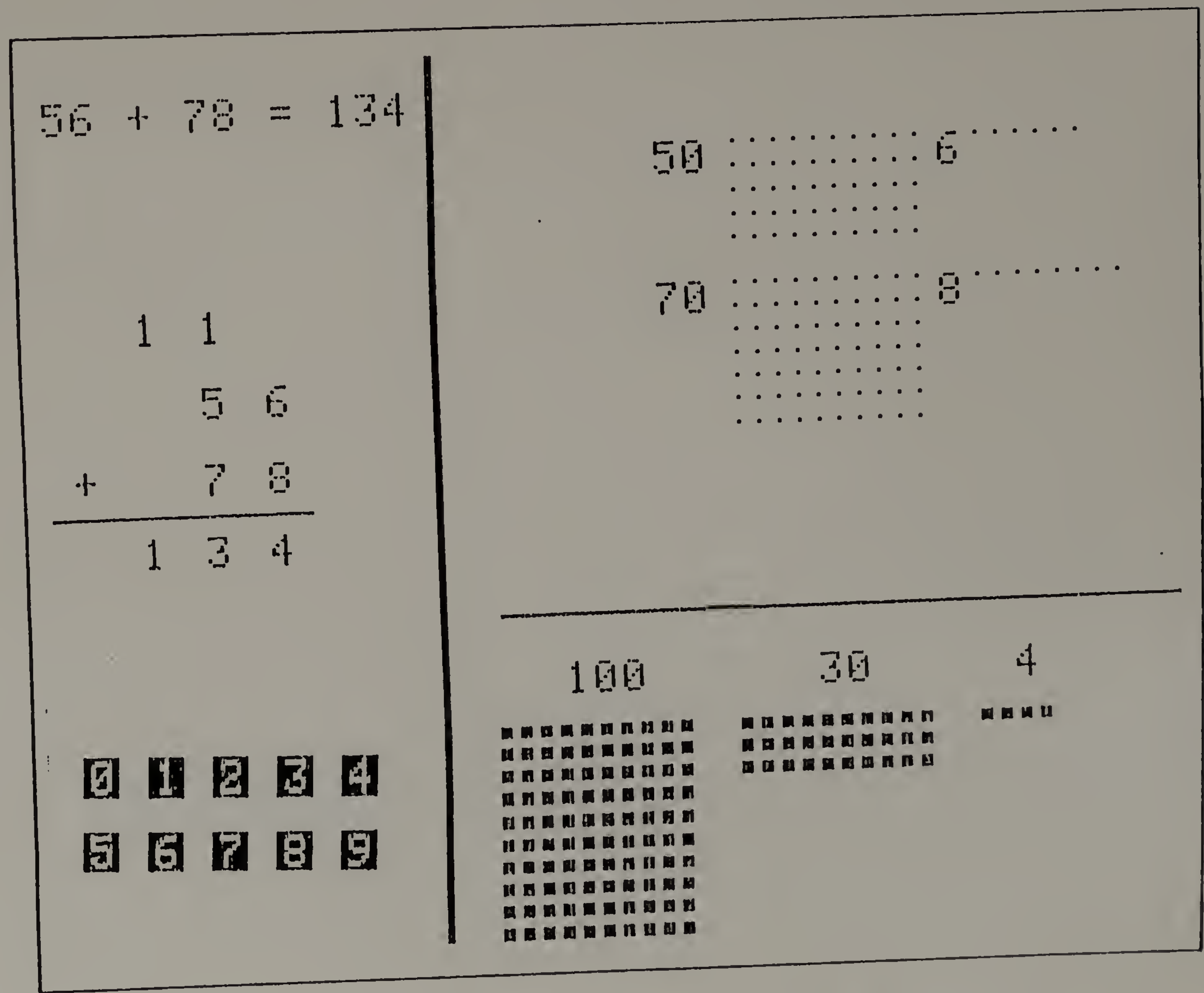

Figure 3.6 Screen display: at end of addition

CHAPTER 4

RESULTS AND ANALYSIS

This chapter contains results and analysis of pretests and posttests, arranged according to basic skills, place value understanding, pencil-on-paper column addition, transfer problems, correction problems, column addition on computer, and timing data. Each section contains both summary statistics and details of performance on components of the various tests.

Data tables for individual subject scores are found in Appendix B.

The three experimental groups differed in the version of software used. The control group had no on-screen number-fact assistance. The assisted group did have on-screen number-fact assistance. The simulation group had not only number-fact assistance but also displays of simulated blocks intended to convey the quantitative meaning of the symbol manipulations in the algorithm as it was executed.

All questions were scored one point for each fully correct answer, zero otherwise.

Although all three groups and almost all subjects improved from pretest to posttest, the matter of primary interest is whether the treatment groups improved to a significantly greater degree than the control group.

Consequently, t-tests were performed on the pretest-to-posttest difference scores. A 5% criterion was adopted for significance, and since the predictions made in the hypotheses being tested are directional, that is, that the treatments will result in improvement, the t-tests were one-tailed. The tests were performed pairwise on the control/assisted and control/simulation groups only. The tests were not done on the assisted/simulation groups because the scores of the simulation group had been predicted to be higher than those of the assisted group when in fact they turned out to be lower. Comparisons were also made on pretest-posttest differences for sex and classroom (Room F vs. Room G).

A further matter of interest was group mean comparisons for tasks that had not been pretested. T-tests were applied pairwise to the control/assisted and control/simulation groups for their performance in basic skills, in the transfer and correction tasks, and in multicolumn addition on the computer.

The following is a brief summary of the analysis.

1. Basic skills scores: no significant differences were found among the groups. This confirms that the composition of the three groups prior to treatment was satisfactorily balanced.

2. Place value pretest/posttest difference scores: both the assisted and simulation groups had significantly higher scores than the control group.

3. Column addition pretest/posttest difference scores: only the assisted group had significantly higher scores.

4. Transfer problems scores: only the assisted group had significantly higher scores.

5. Correction problems scores: only the assisted group had significantly higher scores.

6. Column addition on the computer: no significant differences were found among the groups.

7. Computer timing: the assisted group took significantly less time than either the simulation group or the control group to complete the six problems presented on the computer screen. (See Section 3.3.4 for a description of the light pen. Timing data consisted of the time elapsed between "pen hits". A pen hit occurs when the pen is held stationary for at least 0.5 second against one of the numerals on the screen.)

4.1 Basic Skills

Analysis of basic skills scores is summarized in Table 4.1 below. It indicates that there are no significant differences at the 5% level of significance between control and treatment group scores, between male and female scores, or between Room F and Room G scores.

Discussion of performance on the problems in the basic skills test follows. See Appendix A for details of formation of control and treatment groups.

TABLE 4.1 Basic Skills Scores

Groups	n	Mean	SD	t-value	p
Control	12	21.4	4.9		
Assisted	12	23.4	2.5	1.258	0.111
Control	12	21.4	4.9		
Simulation	12	23.0	3.8	0.882	0.194
Female	20	22.0	3.6		
Male	16	23.4	4.0	1.580	0.178
Room F	21	22.4	3.3		
Room G	15	22.9	4.6	0.332	0.371

Maximum score: 28

Table 4.2 Basic Skills Scores by Problem, on the following page, lists scores and percentages for the entire sample of 36 subjects on each question in the basic skills test. The salient findings in each component of the test are summarized in each of the sections following. Of particular interest in Section 4.8.2 is the listing of the various kinds of errors made in addition tests.

TABLE 4.2 Basic Skills Scores by Problem

		Score (Max: 36)	Percent
Counting			
#1	Count from 14	35	97
#2	Count from 87	32	89
#3	Count by twos	22	61
#4	Count by fives	23	64
#5	Count by tens	33	93
#6	Count by hundreds	7	19
	Total	152	70
Oral simple addition			
#7	4+2	35	97
#8	3+5	33	92
#9	11+6	27	75
#10	7+8	28	78
#11	13+0	36	100
#12	4+6	35	97
#13	6+4	36	100
#14	10+7	28	78
#15	5+50	26	72
	Total	284	88
Written simple addition			
#16	6+3	35	97
#17	2+5	36	100
#18	4+5	34	94
#19	6+3	35	97
#20	3+8	33	92
#21	7+2	34	94
	Total	207	96
Reading 2-, 3-digit numbers			
#22	Read 54	34	94
#23	Read 776	18	50
#24	Read 308	16	44
	Total	68	63
Counting money			
#25	Cents in a dime?	33	92
#26	3 dimes, 7 pennies	25	69
#27	2 dimes, 15 pennies	24	67
#28	Pick out 43 cents	21	58

4.1.1 Counting

Almost all the subjects (97%) successfully counted-on from 14. Slightly fewer (92%) counted-on successfully from 87; of these a few had difficulty crossing 100, either stopping or counting on by tens (110, 120, etc.). Almost all (92%) were able to count by tens but only approximately 60% could count by twos or fives. This difference may reflect the relatively higher frequency of practice in counting by tens in games played outside school. Only one out of five subjects (19%) was able to count by hundreds until prompted ("100, 200 ...?").

4.1.2 Oral/Written Simple Addition

"Simple" addition consisted of addition of two 1-digit numbers or of a 2-digit and 1-digit number. The problems were either presented orally for oral response or presented on separate cards for written response. Overall score for oral simple addition was 88% compared to 96% for written simple addition. Some difficulty was encountered (scores: 72-78%) when the sum of the digits exceeded ten. None of the subjects had completely automated number facts. That is, all subjects resorted to adding by counting, primarily on their fingers (openly or surreptitiously) or occasionally subvocally.

Problems #12 (4+6) and #13 (6+4) were commutation problems with the same addends but in reverse order. Fourteen subjects (39%) commuted; that is, they responded immediately to the second of these problems with the same sum. When asked to explain their rapid response, they said, typically: "You just changed the numbers around", "Same numbers", "[added] backwards", "The same. It didn't change". Twenty-two subjects (61%) did not commute but re-added the numbers, primarily by finger counting.

Problem #15 (5+50) is also a commutation problem. Nine subjects (25%) counted-on from 50 and did not take advantage of the decimal structure of the number system (units nested in tens) or counting by fives. A similar lack of utilizing decimal structure was found with Problem #14 (10+7) which 26 subjects (67%) solved by counting-on from ten.

Problem #11 (13+0) was answered correctly by all subjects. When asked to explain the rapidity of their response, they said typically that zero is "nothing". Here is a sampling of their remarks:

> "No other number."
>
> "You don't add anything."
>
> "Zero is nothing"
>
> "Zero means nothing."
>
> "If you put zero, it's none."

"It's the same."

"Zero doesn't do anything."

"No other number goes with it."

"Zero is hardly a number, just a circle, and a circle is nothing."

These notions conflict with the role of zero as an empty place holder in multidigit numbers and is a source of difficulty in reading and manipulating numbers of three or more digits, of which at least one digit is a zero.

4.1.3 Reading 2- or 3-Digit Numbers

Problems #22-24 required a subject to read a 2- or 3-digit number displayed on a card. There were three numbers to read: 54, 776, and 308. Nearly all subjects (94%) could read 54. Ability to read 3-digit numbers dropped markedly: to 50% able to read 776, and to 44% able to read 308. Many did not respond.

Here is a sampling of misreadings of 776:

"Seven seven six"

"Seventy seven six"

"Seventy six"

"Seventy seventy six"

"One hundred seventy, seventy six"

And here is a sampling of misreadings of 308:

"Three eight"

"Thirty hundred and eight"

"Thirty eight"

"Thirty, eight"

"Thirty eight hundred"

"Three eighty, three hundred thirty eight"

"Three hundred and eighty"

"Three zero seven"

They had been taught to read 3-digit numbers but were not yet sufficiently practiced. They were able pick up the verbal pattern again quickly and could read 3-digit numbers with a little instruction after the completion of the pretest. This deficiency, however, is bound to affect their understanding of place value.

4.1.4 Counting Money

Nearly all subjects (92%) knew there are ten cents "in" a dime. Ten subjects (28%) failed to calculate the value of a random assortment of dimes and pennies correctly or failed to count out a specific amount of money correctly. Counting money is not only a necessary practical skill but also this concrete experience of grouping is a valuable contribution to place value understanding.

4.2 Place Value

Pretest/posttest place value mean scores are summarized in Table 4.3 below. Each test question answered correctly was scored one point, zero otherwise.

TABLE 4.3 Mean Pretest-Posttest Place Value Scores

Group	Pretest	Posttest	Difference
Control	6.6	11.4	4.8
Assisted	5.9	14.5	8.6
Simulation	6.3	14.7	8.3

Maximum score: 21

An analysis of place value difference scores shown in Table 4.4, on the following page, indicates significantly higher scores for both the assisted and simulation groups, but no significant differences in the sex and classroom comparisons.

TABLE 4.4 Place Value Difference Scores

Group	n	Mean	SD	t-value	p
Control	12	4.8	2.9		
Assisted	12	8.6	3.3	2.969	.004 *
Control	12	4.8	2.9		
Simulation	12	8.3	3.9	2.492	.011 *
Assisted	12	8.6	3.3		
Simulation	12	8.3	3.9	.171	.866
Female	20	6.7	4.2		
Male	16	8.0	3.1	1.087	.143
Room F	21	7.1	4.0		
Room G	15	7.4	3.4	0.202	.421

* Significant at the 5% level

Table 4.5 Place Value Scores by Problem, on the following page, lists scores and percentages for the entire sample of 36 subjects on each question in the place value tests (pretest/posttest). Each test question answered correctly was scored one point, zero otherwise.

The table is not broken into control and treatment groups since the focus of interest is on the relative difficulty of the problems posed. The salient findings in each component of the test are summarized in each of the sections following the table.

TABLE 4.5 Place Value Scores by Problem

		Scores (Max:36)		Percent	
		Pretest	Posttest	Pretest	Posttest
Question Type					
What does the digit mean?					
#1	54	5	29	19	81
#2	54	9	36	25	100
#3	776	5	32	14	89
#4	776	5	25	14	69
#5	776	9	30	25	83
#6	308	9	31	25	86
#7	308	8	25	22	69
#8	308	4	20	11	56
	Total	54	228	19	79
Which number is larger?					
#9	522 vs 288	27	28	75	78
#10	799 vs 877	25	28	69	78
	Total	52	56	72	78
How many tens/hundreds in...?					
#11	146	5	17	14	47
#12	52	19	28	53	78
#13	378	12	20	33	56
#14	529	14	21	39	58
	Total	50	86	35	60
Name tens					
#15	Four tens	17	22	47	61
#16	Ten tens	4	28	11	78
	Total	21	50	29	69
Positional value					
#17		7	18	19	50
Decomposition					
#18		0	3	0	8
Composition					
#19		6	8	17	22
Proximity: 1-digit numbers					
#20		30	28	83	78
Proximity: 3-digit numbers					
#21		6	10	17	28
Overall Totals		226	487	30	64

4.2.1 Digit Meaning

For Problems #1 through #8 that ask for the place value of a digit, the pretest score for all subjects was 19% compared to the posttest score of 79% The low pretest score indicated that few subjects had been schooled sufficiently in place value or even understood the problem. Many subjects did not respond in the pretest. Posttesting suggested lingering difficulty with the hundreds place and particularly with the zero in 308. Here is a sampling of failed responses:

Gives any digit a tens value.

Responds with the digit plus one.

Guesses(?) "ones", "tens", ignores hundreds place

All digits are given a units value.

Re-reads the number.

Responds with "the ones side" or "the twos side".

The sevens in 776 are called "seven pennies".

> In posttest, this subject responded to the seven in the hundreds place as "one hundred sevens", which is not incorrect but misses the decade structure. To another subject this seven was interpreted as "one hundred seventy".

4.2.2 The Larger of a Number Pair

Problem #9 asked which number is larger, 522 or 288? and Problem #10 asked which number is larger, 799 or 877? There was little pretest to posttest improvement in scores (72% to 78%). Most subjects selected the larger number by mechanically comparing the leftmost digit in each number, but only a few explained their choice by the place value of that digit. Most of those who erred made their choice by selecting the number containing the largest digit of either pair and ignoring place value.

4.2.3 How Many Tens/Hundreds?

Problems #11 through #14, "How many tens/hundreds are there in [number]" are the counterparts of the digit-meaning questions, "What does the [digit] in [the number] mean?" The responses and difficulties were similar. Problem #10 ("How many tens in 146?") was the most difficult of the four with scores of 14% in pretest and 47% in posttest. A response of either "four" or "fourteen" was scored correct. The many (approx. half) who failed Problem #13 ("How many hundreds are there in <u>three</u> hundred ... [pause with emphasis] ... seventy eight?") missed what was offered as a seemingly clear and loud hint in the phrasing and intonation of the number.

4.2.4 Name Tens

Problem #15 ("What are four tens called?") and Problem #16 ("What are ten tens called?") are questions of multiplication before this topic is formally introduced. Knowledge of this is required for an understanding of a carry, when the sum of the tens column exceeds 9. The "1" carried stands for one one-hundred (ten tens) carried into the hundreds column. Few subjects (11%) in the pretest knew that ten tens are called one hundred. A common response was to add ten plus ten, yielding "twenty". Ten out of the 36 subjects (28%) responded this way. In posttest, 78% answered this question correctly, which for at least some of them is quite likely a memorized response to instruction and not an understanding of the re-grouping operation.

4.2.5 Positional Value of a Digit

Problem #17 ("What is the difference between the three in 32 and the three in 73?") yielded a pretest score of 19% and a posttest score of 50% A response technically correct but without reference to place value was graded incorrect, such as, typically, "This three is in front and this other three is in back." Other variations: "first/last", "beginning/end", "not the same".

4.2.6 Decomposition of Numbers

Problem #18 ("What three numbers add up easily to 658?" -- the number was displayed on a card.) was not answered correctly by any of the subjects in pretest, and only 8% did so in posttest. This question can of course be answered any number of ways. It is perhaps too difficult for children at this grade level but still is indicative of place value understanding. It did prove to be the most difficult of all the place value questions. Even a definite hint was not picked up. The hint resided in the way the number was spoken, its emphasis and pauses: "Six hundred ... [pause] ... fifty ... [pause] ... eight". The subjects did not yet have a sufficient understanding of the nested decimal structure of the number system.

4.2.7 Composition of Numbers

Problem #19 ("Can you add up these numbers in your head?" [4+70+200] displayed on a card) is the counterpart to Problem #18. The numbers are presented in this particular order to avoid priming the subject when the problem is spoken aloud. Again, few subjects scored well: 17% in pretest, 22% in posttest.

Here is a sampling of incorrect responses: 247, 201, 1200, 1120, 1300, 294, 301, 904. These are not random responses since the effort to add the numbers can be seen in each answer. Each could be described as a fleeting product of thoughtful guessing, an effort to bring the fragments of one's knowledge to bear on an unfamiliar problem.

4.2.8 Number Proximity

Problem #20 (Which pair of numbers have numbers that are closer together? --- 2,7 (displayed on one card) or 4,5 (displayed together on another card) was answered and explained correctly by most subjects, 83% in pretest, 78% in posttest. Five of the subjects (14%) responded to the word "closer" as a physical attribute rather than as a comparison of number magnitudes. They seemed to be stuck in this interpretation, in spite of prompting to see the numbers as "numbers" (abstractions). They might have responded differently if the question had been put in the context of comparing two sets of ages.

Problem #21, in which the number pairs were 436,448 and 546,548, was much more difficult, resulting in scores of 17% in pretest and 28% in posttest. This reflected the subjects' difficulties with place value in 3-digit numbers.

4.3 Multicolumn Addition

This section contains results of the eight pretest/ posttest multicolumn addition problems. Statistical analysis of the performance of the treatment groups vs. the control group is followed by more detailed results on individual problems. These results are not broken into control and treatment groups since the focus of interest is on the relative difficulty of the problems posed. Errors in column addition, because of their importance to the discussion in Chapter 5, are listed separately in some detail in Section 4.8.2.

Pretest/posttest column addition mean scores are summarized in Table 4.6 below.

TABLE 4.6 Mean Pretest-Posttest Addition Scores

Group	Pretest	Posttest	Difference
Control	2.7	4.0	1.3
Assisted	2.7	6.0	3.3
Simulation	2.3	4.3	2.0

Maximum score: 8

An analysis of column addition difference scores shown in Table 4.7 below indicated significance for the control/assisted group only, none for any others.

TABLE 4.7 Column Addition Difference Scores

Group	n	Mean	SD	t-value	p
Control	12	1.3	2.7		
Assisted	12	3.3	2.0	2.036	.027 *
Control	12	1.3	2.7		
Simulation	12	2.0	2.0	0.675	.254
Assisted	12	3.3	2.0		
Simulation	12	2.0	2.0	0.675	.122
Female	20	2.4	2.5		
Male	16	2.1	2.3	0.356	.362
Room F	21	2.4	2.7		
Room G	15	1.9	1.9	0.610	.273

* Significant at the 5% level

The pretest/posttest 2-digit addition problems were a mix of seven of the nine types. Overall performance improved from 32% in pretest to 60% in posttest. See Table 4.8, Column Addition by Problem, below.

TABLE 4.8 Column Addition by Problem

			Scores (Max: 36)		Percent	
Problem		Type	Pretest	Posttest	Pretest	Posttest
#1	45+3	3	29	30	81	83
#2	13+46	6	20	29	56	81
#3	88+37	9	0	19	0	53
#4	96+7	5	14	20	39	56
#5	5+68	4	21	18	58	50
#6	26+38	7	4	21	11	58
#7	54+62	8	4	21	11	58
#8	84+67	9	0	14	0	39
Total			92	172	32	60

Problem #1 (45+3) scored highest (81% in pretest, 83% in posttest) since most subjects simply counted-on 3 units without resorting to column addition. Problem #2 (13+46), which involved no carries and could also be solved by counting-on from 46, scored next highest (56% in pretest, 81% in posttest).

The lowest scores occurred in pretest, as expected, since instruction in multicolumn addition with carries had not yet begun. None of the subjects in pretest correctly solved the two type 9 problems (#3, #8), which involved two carries.

All eight problems were presented to the subjects in horizontal format since an instructional objective was to learn that multidigit numbers presented in horizontal format should be rewritten in vertical format ("up and down") for easier solution using the algorithm. Pretest indicated that nearly all subjects (92%) tried to solve many problems in horizontal format. In posttest this percentage dropped to 53%, which is still relatively high. Many subjects persisted in trying to solve the more difficult problems in the horizontal format in which the problems were presented, even after instruction.

Problems #1, #7, and possibly #2 are easily solved by counting-on from the larger addend. In pretest most subjects did this as expected, but also, lacking knowledge of the algorithm, they tried to solve some of the more difficult problems by counting-on. The percent subjects counting-on dropped from 86% in pretest to 33% in posttest.

In pretest, 39% of the subjects did not respond to (left blank) one or more problems, but in posttest all subjects responded to all the problems.

The "monitor" test of three problems, one each of types 7, 8, and 9, indicated that some learning of the algorithm had occurred during the several weeks time that elapsed between the pretest and the beginning of instruction. Performance score on these three problems was 25%, which is greater than the pretest score of 7% on comparable type problems, but less than the posttest score of 54% on comparable type problems.

4.4 Transfer Problems

The transfer problems, so called, were administered to the subjects immediately following the posttest column addition problems. They were intended to elicit a transfer of the subject's newly acquired algorithmic skill to solve more complex column addition problems which had not yet been encountered in the classroom. The problem set consisted of six transfer problems, two each of the following problem types to add: three 2-digit numbers, two 3-digit numbers, and two 4-digit numbers.

Table 4.9 on the following page, Mean Scores for Transfer Problems, indicates a significant difference in means between the control/assisted groups (one-tailed t-test) and between the assisted/simulation groups (two-tailed t-test). Table 4.10 following, lists results for individual transfer problems.

TABLE 4.9 Mean Scores for Transfer Problems

Group	n	Mean	SD	t-value	p
Control	12	2.2	1.9		
Assisted	12	4.1	1.9	2.423	.012 *
Control	12	2.2	1.9		
Simulation	12	2.4	1.9	0.316	.378
Assisted	12	4.1	1.9		
Simulation	12	2.4	1.9	2.117	.046 *

* Significant at the 5% level

Maximum score: 6

TABLE 4.10 Scores for Individual Transfer Problems

Problem		Score (Max: 36)	Percent
#1	68+42+57	18	50
#2	79+37+16	10	28
#3	407+847	15	42
#4	977+221	26	72
#5	2847+3625	16	44
#6	5474+4378	19	53
	Total	104	48

Would subjects, having been instructed with examples requiring carries of "one", transfer their knowledge and understanding of the algorithm to a problem requiring a carry of "two"? Transfer Problems #1 and #2 (scored 46%) compared to addition posttest Problems #3 and #8 (scored 39%), which are comparable in type, indicates that some transfer has occurred. However, transfer Problem #2, which entailed a carry of "2", scored low (28%), primarily because several subjects either did not sum all three digits in the units column or believed that "1" is always carried.

Transfer Problems #3 through #6 tested the subjects' ability to extend the algorithm to 3- and 4-digit addends. The composite score of these four problems was 53%. The composite score of the four problems of comparable type (involving two carries) in the posttest addition Problems #3, #6, #7, #8, was 52%, which also indicates that transfer of skill occurred.

Being challenged with novel and more difficult problems seemed to disconcert several subjects, who apparently regressed into making errors that were less prevalent or even not seen in the addition posttest. Table 4.11, Incidence of Errors (Posttest vs. Combined Transfer and Correction Problems) indicates that two types of procedural error more than doubled in incidence from the posttest problems to the transfer and correction problems. The two types of error were neglecting-to-carry (e.g., 27+36 = 53) and carry-into-answer (e.g., 27+36 = 513). Also, when more digits are to be added, as expected, the incidence of miscalculation (number-fact error) increased.

TABLE 4.11 Incidence of Errors *

Posttest vs. Combined Transfer and Correction Problems

	Posttest Problems		Combined transfer and correction Problems	
	Incidents	Incidence	Incidents	Incidence
Miscalculations	19	6.6	41	11.4
Procedural errors				
No response	0	0.0	16	4.4
Did not carry	11	3.8	39	10.8
Wrong carry	0	0.0	6	1.7
Carry in answer	13	4.5	47	13.1
Undeterm. error	18	6.2	20	5.6
Add col. left-rt	16	5.6	10	2.8
Incomplete	1	0.3	6	1.7
Carried left-rt	8	2.8	15	4.2

* Note: Posttest incidence base is 36 subjects working 8 problems. Combined transfer and correction incidence base is 36 subjects working 10 problems. Incidence is defined here as incidents per 100 problems.

4.5 Correction Problems

The correction problems, so called, are a set of four completed but erroneous column addition problems administered to the subjects immediately following the transfer problems. The subjects' task was to detect the errors, correct them, and articulate reasons for doing so.

Table 4.12 below, Mean Scores for Correction Problems, indicates a significant difference in means between the control/assisted groups (one-tailed t-test) and between the assisted/simulation groups (two-tailed t-test).

TABLE 4.12 Mean Scores for Correction Problems

Group	n	Mean	SD	t-value	p
Control	12	1.5	1.6		
Assisted	12	2.7	1.4	1.902	.035 *
Control	12	1.5	1.6		
Simulation	12	1.4	1.4	0.136	.447
Assisted	12	2.7	1.4		
Simulation	12	1.4	1.4	2.227	.036 *

* Significant at the 5% level

Maximum score: 4

Table 4.13 below lists the scores for individual correction problems. A correction problem was graded incorrect if the subject incorrectly solved the problem or judged the solution to an erroneous problem to be correct. Most subjects did not articulate reasons for making or not making corrections. Either unwilling or unable to interpret the answer given, they proceeded to do the algorithm and superimposed their own answer or not, without comment.

TABLE 4.13 Scores for Individual Correction Problems

Problem and its error			Score (Max: 36)	Percent
#1	54+38=93	Miscalculation	19	53
#2	26+18=314	Carry-in-answer	11	31
#3	46+37=73	Did not carry	16	44
#4	29+1 =20	Did not carry	21	58
		Total	67	47

4.6 Multicolumn Addition on Computer

Of the 30-problem set used in the instruction phase, the last six were used as a test of performance of the standard addition algorithm as it was learned on the computer by means of the light pen.

Table 4.14, Column Addition on the Computer, on the following page contains the results and t-test analysis of three aspects of this test: mean scores, "significant" errors, and "thrashing" errors.

"Significant" errors, in the context of performing the computer algorithm with the light pen, are those which if performed in a pencil-on-paper test would have been scored incorrect (e.g. incorrect 1-digit addition or misplacing the carry from the tens column into the units column).

"Thrashing" errors are those which are recorded by the software as "mis-hits" but would not have been scored incorrect in a pencil-on-paper test (e.g. holding the light pen too long in one position or placing the hundreds carry too high in the hundreds column).

No significant differences were found in any of the group comparisons.

TABLE 4.14 Column Addition on the Computer

Group	n	Mean	SD	t-value	p
Mean Scores (Max. 6)					
Control	12	3.8	1.7		
Assisted	12	3.8	1.4	0.130	.449
Control	12	3.8	1.7		
Simulation	12	3.7	1.6	0.123	.452
Significant Errors (Means)					
Control	12	4.0	4.6		
Assisted	12	3.3	3.1	0.473	.321
Control	12	4.0	4.6		
Simulation	12	3.7	4.1	0.188	.452
Thrashing Errors (Means)					
Control	12	15.7	12.9		
Assisted	12	12.8	9.3	0.635	.266
Control	12	15.7	12.9		
Simulation	12	13.3	12.2	0.454	.327

4.7 Timing

During the 6-problem test of performing the algorithm on computer, the dwell time (seconds) between pen hits as each subject performed the steps in the algorithm was captured by the software. These times were compiled into three summary statistics: mean time (sec) in correct moves, total time (sec), and percent time in correct moves. All of these times refer to the six computer test problems. The results and analysis are found in Table 4.15 Mean Computer Times, on the following page .

Mean time in making correct moves and mean total time were found to be significantly less for the assisted group than either for the control or simulation groups.

No significant differences were found among any of the groups for percent time in making correct moves. That is, all three groups consumed about 20% of their total time in making errors, but the assisted group was significantly faster overall than either of the other two groups.

TABLE 4.15 Mean Computer Times

Group	n	Mean	SD	t-value	p
Time (sec) in Correct Moves					
Control	12	261	67		
Assisted	12	220	34	1.882	.037 *
Control	12	261	67		
Simulation	12	231	54	1.201	.122
Total Time (sec)					
Control	12	332	114		
Assisted	12	267	61	1.724	.049 *
Control	12	332	114		
Simulation	12	306	133	0.516	.306
Percent Time in Correct Moves					
Control	12	82.1	14.2		
Assisted	12	83.8	8.4	0.369	.358
Control	12	82.1	14.2		
Simulation	12	80.2	13.0	0.345	.377

* Significant at the 5% level

4.8 Errors in Multicolumn Addition

The frequencies and kinds of errors in multicolumn addition made by the second-graders in this study are set out in some detail in this section. A study of errors has always been of great interest in educational research for suggesting insights into children's thinking and behavior. In the next chapter this becomes an important basis of discussion and interpretation.

4.8.1 Frequencies

First, for an overview, Table 4.16 below compares the frequency of procedural versus calculation errors among the 36 children solving the eight addition problems in pretest and posttest. Procedural errors, those that involve placement of the numbers, predominated over calculation errors, those that involve obtaining the numbers to be placed. The category "No response" could be classified as a form of procedural error: being unable or unwilling to proceed. Calculation error includes mistaken recall of number-facts and/or a mis-reconstruction of number-facts by counting.

TABLE 4.16 Procedural vs Calculation Errors

	Pretest		Posttest	
	Frequency	Percent	Frequency	Percent
Calculation errors	21	13	19	16
Procedural errors	91	56	102	84
No response	49	31	0	0
Total	161		121	

Another way of looking at error frequencies (see Table 4.17 below) is to count the number of subjects out of the whole sample of 36 making a particular kind of error.

TABLE 4.17 Frequency of Procedural Errors

	Pretest		Posttest	
Type of error	Number of Subjects	Percent	Number of Subjects	Percent
Did not carry	5	14	4	11
Misalign digits	3	8	14	39
Carry-in-answer	4	11	5	14
Response undeterm.	19	53	5	14
Add col. left-right	1	3	4	11
Incomplete	4	11	1	3
Carry left-right	2	6	4	11
Add horiz incorrectly	6	17	3	8

The subjects changed their approach to the addition problems from pretest to posttest. In pretest almost all (33 or 92%) tried to solve at least some multidigit addition problems in horizontal format. Many subjects seemed to be constrained by the format of the problem, trying to solve a horizontally formatted problem without rewriting it into vertical format for algorithmic treatment. This may indicate a reliance on their familiar counting-on method of addition. It may also indicate habituation to work-book formats: a horizontally presented problem is to be solved horizontally. The number of such subjects declined to 19 (53%) in posttest after instruction in multicolumn addition. A large proportion of subjects left at least one problem blank (no response) in pretest, but none did so in posttest. Nearly half of the subjects (15 or 42%), the same number in both pretest and posttest, made at least one calculation error in the eight problems.

4.8.2 Individual Error Types

The following is a collection of nearly fifty multidigit addition errors, almost all different to some degree, and all drawn from the sample of 36 second-graders. Given a larger sample, more types of error are likely to be found. The errors have been grouped according to the subject making them and have been labelled with the subject's initials. This has been done to indicate the numerous instances of knowledge instability in which the same kind of problem is solved in different buggy ways by the same subject. Also recall that all of the problems are presented in horizontal format and that many subjects chose or felt constrained to solve them without rewriting them vertically.

Note the general characteristics of the errors, which may be seen as the result of knowledge that is missing or incomplete, fragmentary, inappropriately applied (buggy), unstable, and "set" or entrenched. These characteristics and the many following instances are used in the next chapter to develop a comprehensive approach towards all these disparate findings.

1. AD Carry in answer. (Is this a miscalculation or a carry in each column?) 88+37 -> 1216

2. AD Misaligned. Rewrote (5+68) as:

```
  5
 68
---
118
```

3. TAN Added incorrect pairs of digits:
13+46 -> (1+4),([3+4] or [1+6]) -> 57

4. TAN Added all digits: 96+7 -> (9+6+7) -> 22

5. TAN The plus symbol displaced the second addend to the right. Carry-in-answer.
Rewrote 56+78 as:

```
56
+78
----
1216
```

6. TAN Said "One hundred and three" but writes "1300":
96+7 -> 1300

7. TAN Added columns left to right, perceiving that 5,6 are to be added but stopped (because the columns in the answer are occupied?). Rewrote 54+62 as:

```
 54
+62
---
 11
```

8. JES Broke up second addend into separate digits to be added: 13+46 -> (13+4+6) -> 22

9. JES Broke a 2-digit number into separate digits, addedthem in pairs and reassembled the results into a 2-digit number:
54+62 -> (5+4),4+2) -> 96

10. JES The many zeros indicate an indefinitely large number beyond 100: 84+67 -> 10000

11. JES Dropped both carries. Rewrote 56+78 as:

```
 56
+78
 24
```

12. HC Counted on from 37? 88+37 -> (37+8+8) -> 44

13. HC Tried to write one hundred three: 96+7 -> 300

14. HC Cross-added digits. Carry-in-answer: 84+67 -> (8+7),(4+6) -> 1510

15. DF Ignored one addend. Added digits 4,2: 86+42 -> 6

16. DF Both the problem and its addends were rewritten vertically: (26+38):

```
 23
+68
 91
```

17. GM A mix of proper carry and carry-in-answer:

```
   1
 2847
 3625
51472
```

18. FR Added columns right to left, carried from tens column into the units column. Then re-added the units column, scratching out the original 8 and replacing it with 9. Rewrote (86+42) as:

```
  1
 86
+42
 28
  9
```

19. FR Dyslexic reversal? Added left to right putting carry from the hundreds column into the tens column. Finally, added the units column and put the carry into the answer (as when adding columns right to left):

```
    1
 2847
+3625
54712
```

20. JL Cross-added digits: 13+46 -> (1+6),(3+4) -> 77

21. JL Wrote 103 literally as "1003": 96+7 -> 1003

22. ER Added units then tens, but placed partial sums in reverse order, from left to right:

26+38 -> 145

23. ER Mixed addition and subtraction, subtraction in the units column, addition in the tens column:

```
 416
  5 6
 +7 8
 11 8
```

24. ER Scrambled carry. Intended to carry 20 but put 2 in the units answer place and carried the zero into the tens column:

```
  0
 79
 37
+16
112
```

25. MIS Mixed horizontal and vertical procedures, reversed answer digits. Added 6 and 8, put 4 in the answer and carried "1" over the 2; added 1+2+3, put a six to the right of the four in the answer:

```
1
26+38 -> 46
```

26. BED Carried a ten by encircling the 1, possibly as a reminder that a ten is being carried.

27. BED Omitted the zero in the tens place. (96+7) was rewritten as:

```
 1
96
 7
13
```

28. JSS Added and carried left to right:

```
  1
 86
+42
 29
```

29. JSS Added 1-digit addend twice:

```
 45
+ 3
 78
```

30. JSS Was observed to reverse the digits in a 2-digit number (15 was written as 51):

```
  1
 84
+67
511
```

31. CG Inability to write a number over 100. CG wrote "3" and then said "This is one hundred three".

96+7 -> 3

32. KEF Solved all posttest problems in both horizontal and vertical format, getting different answers, unaware of or ignoring any inconsistency. Note the way the digits are vertically formatted without regard to their values, although the correct procedure was carried out. The problem presented was (5+68) and was solved in two ways:

```
                  1
5+68 -> 73       56
                + 8
                 64
```

33. KH Was observed to put the tens carry back into the tens column. Rewrote (54+62) as:

```
 1
 54
+62
 16
```

34. EL Apparently counted on from the larger addend but ignored the tens digit on the other addend:

26+38 -> 38+6 -> 44

35. IA Misaligned but correct. Rewrote (26+38) as:

```
1
26
+38
 64
```

36. IA Now aligned correctly. Added right to left but carried back into the units column. Then re-added the units column, scratching out the previous answer. Rewrote 54+62) as:

```
  1
 54
+62
 16
  7
```

37. IA Cycling. Similar to (36) above. Added units column and carried into the tens column. Added tens column but either ignored the carry or miscalculated 8+6 to be 14. The carry from the tens column is then put into the units column, which is then recalculated and the answer changed to 2, scratching out the previous answer. The problem (84+67) was rewritten as:

```
 11
 84
+67
 41
  2
```

38. AL Digits were added correctly in parallel and then inappropriately combined:

56+78 -> (5+7)+(6+8) -> (12+14) -> 26

39. BG Digits as well as numbers were formatted vertically: (13+46) was rewritten as:

```
 3 6
+1+4
 410
```

40. VR Left to right column addition, carry plays no role. Added tens column first, then the units column, putting the carry into the tens column. The problem (26+38) was rewritten as:

```
 1
 26
+38
 54
```

41. VR Left to right addition and carry. The problem (54+62) was rewritten as: 1

```
 54
+62
 17
```

42. VR Cycling. Solved (84+67) as follows:
Added 4 and 7, put result (11) in answer. Remembered to put carry over tens column, erased the 1 in the tens answer place. Added tens column, put 5 in the tens column. Carried into the units column without changing the answer in the units column.

```
 11
 84
+67
 11
 5
```

The next three problems were done by JK in succession.

43. JK Cycling. Carried into the units column. Added tens. Noticed carry in units column, so re-added units column. (26+38) was rewritten as:

```
  1
 26
+38
 54
  5
```

44. JK Ignored carry or forgot to carry to the hundreds place. (54+62) was rewritten as:

```
 54
+62
 16
```

45. JK Carry-in-answer. Rewrote (84+67) as:

```
 84
+67
1411
```

46. JK Added units column, putting 5 in units answer place and putting the carry above 8 in the <u>units</u> column. Added tens, putting the carry in the <u>tens</u> column. Added up the tens column (result 12). Then put 12 in the tens answer place. Re-added the units column (result 16). Put 6 next to the units answer without deleting the 5.
Rewrote (88+37) as:

```
 11
 88
+37
1256
```

47. JK Similar to (46) above, but now put a carry in the hundreds column:

```
111
 56
+78
124
  5
```

48. JK Observed to first add left to right the "easy" digits, those that sum to less than 10, then the "hard" ones right to left, squeezing the carries into the answer:

```
  2 84 7
+ 3 62 5
  514612
```

50. DB "You can put the 1 anywhere", said DB. Db put the carries from the units and tens columns off to the left. The leftmost column was then summed to 20: (11+5+4 -> 20):

```
11
   5474
  +4378
  20742
```

Yet this subject scored very well, ranking second in the place value posttest. DB answered 19 of the 21 problems correctly, an indication that test scores used alone for diagnosis may obscure fundamental deficiencies in understanding.

CHAPTER 5

DISCUSSION AND RECOMMENDATIONS

This study has focused on two aspects of algorithmic learning: the child's short term processing capacity and place value understanding. Section 5.1 discusses these in terms of the analysis of data reported in Chapter 4.

Collected along with pretest/posttest performance scores were qualitative data regarding the kinds of errors made. The diversity of errors and the evidence they provide about children's mathematical knowledge also demand interpretation. Consequently, in Sections 5.2 and 5.3 an interpretive and diagnostic framework for children's errors in elementary mathematics is developed in the form of a metaphor I have called "superposition of frames". Section 5.4 discusses the educational implications of the study and its interpretations. Finally, Section 5.5 consists of a brief retrospective summary of the study.

5.1 Interpretation of Results

How do the results reported in Chapter 4 bear on the hypotheses posed at the beginning of this study? Figure 5.1 below displays mean pretest-posttest difference scores, the key measures in the place value and addition tests to be discussed in this section.

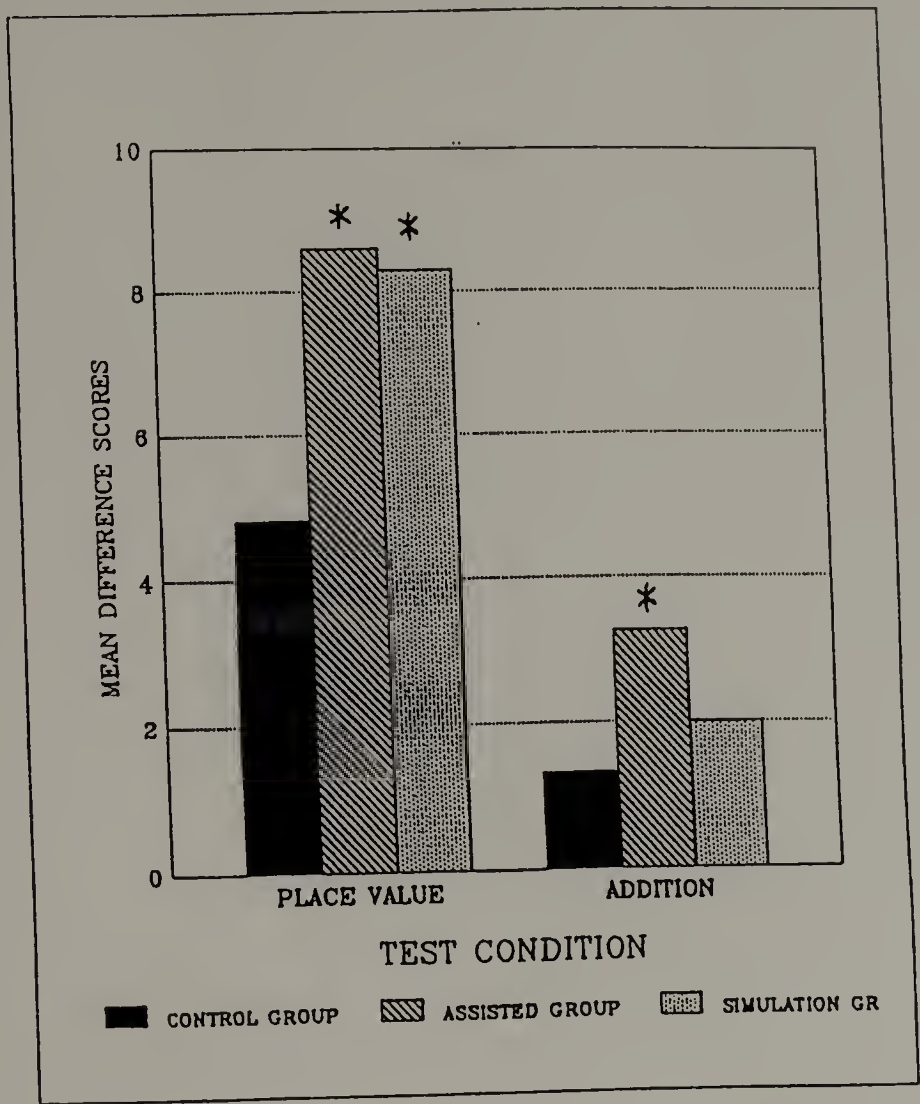

FIGURE 5.1 Mean pretest-posttest difference scores.

* Difference significant at the 5% level when compared to pretest/posttest difference of the control group.

The first hypothesis --- that multicolumn addition is learned more effectively when learned first as a sequence of procedural steps alone and without initial recall of number facts than when the algorithm is learned along with required recall of number facts --- is supported, but not unequivocally, by the pretest-posttest difference scores. Both assisted and simulation groups had significantly higher scores than the control group in the place value posttest, but in the multicolumn addition posttest only the assisted group (but not the simulation group) had significantly higher scores than the control group. Again, only the assisted group had significantly higher scores in the transfer and correction problems (see Table 4.9 and Table 4.12), which were also pencil-on-paper multicolumn addition tests, An anomaly (to be explained below) is the finding that although the simulation group did not score significantly higher than the control group, except for the place value posttest, yet it too, like the assisted group, had received on screen number-fact assistance.

The second hypothesis -- that simulating the movements and quantitative meaning of the symbol manipulations by means of a simultaneous display of graphic blocks on the computer screen would result in more effective learning of the algorithm by the simulation group than either the assisted group or the control group -- was not supported by the data. The simulation group did not score higher than the assisted group in any of the posttests and scored significantly higher than the control test only in the place value posttest.

An explanation for this apparent anomaly may lie in what was claimed in the first hypothesis, for which some support was found. If, in fact, the processing capacity of a subject in the control group is exceeded by having to recall or reconstruct number facts while learning the steps of the algorithm, then the <u>reduction in demand</u> made on the simulation group by the contribution of on-screen number-fact assistance in learning the algorithm is replaced by or offset by the <u>greater demand</u> of the complex workings of the simulated blocks and the accompanying instructions. In effect, the benefit of the on-screen assistance is negated by the competing stimuli of the simulation displays. This benefit apparently is not negated when the simulation group is learning place value. Here, the significant increase in place value scores of the simulation group may be attributed to the of simulation displays and the accompanying instruction. The assisted group did not have these displays but still scored significantly higher than the control group in the place value posttest. This may be attributed to the processing capacity made available by on-screen number-fact assistance, consequently a greater possibility of attending to and processing instruction on place value.

Another anomaly to be explained is the contrast in performance of the three groups on the addition algorithm done with the light pen on the computer screen (which will be referred to in later discussion as the computer addition) versus the pencil-on-paper addition algorithm. There were no significant differences found among the three groups on the computer addition, yet the assisted group scored significantly higher than either the control or simulation groups on the pencil-on-paper addition posttest. Ostensibly the two kinds of addition, whether on the screen or on paper, have many similarities. They use the same symbols; the symbol manipulations for the most part are the same; the light pen is closely analogous to an ordinary pen, etc. Yet there are differences from a pen primarily in that answer numbers or carries need to be fetched from the 0-9 array and that the subject cannot literally write with the light pen, etc. Many subjects did not see the connection between the computer addition and pencil-on-paper addition. What they learned by doing the computer addition did not transfer to pencil-on-paper addition, where they reverted to the buggy algorithms seen in their pretests.

To summarize:

First, evidence has been found that learning multicolumn addition by means of the software developed for this study is more effective when on-screen number-fact assistance is provided. Effective learning is expressed here in terms of significantly higher pretest-posttest difference scores in a pencil-on-paper addition test and in a test of place value understanding. The general conclusion is that an algorithm is learned more effectively if some of the demand on short term memory is temporarily lifted, such as the child's effort to recall or reconstruct number facts or the instructor's imposition of explanatory material.

Second, the version of the software designed to enhance place value understanding by simultaneously displaying simulated blocks which mimic the symbol manipulations of the algorithm, was found to be only partially effective. Significantly higher scores occurred in place value understanding but not in algorithmic performance.

However, there may be other explanations or variables contributing to the differences found in group performance and to the anomalous results described above. Some of these other explanations or variables may be:

1. Instruction provided during treatment may not have been sufficiently uniform throughout the three groups. This is possible since each subject was individually instructed; however, there is no overt indication of a significant change in instruction over the course of the study. If instruction had changed significantly, for example, if it had improved, we would expect to see scores correlated positively with subjects starting instructional treatment at later dates. But subjects' starting dates were randomized and there is no correlation between posttest scores and the time when posttests were given. Of course, since the block simulations required explanation, instruction of the simulation group had to be different and longer than that of the other groups.

2. Subjects with high memory and/or attention skills or those having perceptual-motor difficulties may not have been equally distributed among the three groups. There was no testing of these skills and abilities; however, it is a reasonable assumption that they correlate highly with high pretest scores which were fairly represented in all three groups.

3. The amount and quality of assistance that subjects may have been getting at home during treatment may not be equally distributed among the three groups. This was not determined and remains an open question.

4. The number and intensity of distractions occurring in the vicinity of the table where the subjects were tested was not controlled. There was no separate or private room in which the study might be conducted. The only site made available to the researcher was a table in the school's large central open area also used by reading groups and the school library. There were frequent groups of children and of visitors passing nearby. Some children were more easily distracted than others.

5. There was limited transfer of what was learned about the computer-based algorithm to its pencil-on-paper implementation, which is the basis of the posttest. Strong evidence that this occurred is indicated in an examination of the kinds of errors made. (See Section 4.8.2 Individual Error Types).

6. High scores in the place value posttest alone do not necessarily imply understanding. Responses in this test may be rote and a reflection of an ability to recall specific instructions and explanations without fully understanding the implications of what is recalled. Again, an examination of the kinds of errors lends support to this effect.

7. The relatively small sample size, 12 subjects in each of 3 groups, may have been insufficient to fully bring out other effects that reach statistical significance.

8. There were only two instructional sessions. One or two more sessions might have elicited stronger effects, particularly between the assisted and simulation groups.

5.2 Developing a Perspective on the Data

What accounts for the apparent anomalies in the analysis of the data discussed in the preceding section? Why did many children revert back to an idiosyncratic, buggy performance in the pencil-on-paper posttest and not transfer procedural skill acquired on the computer? Why did they feel constrained to solve difficult multidigit addition problems without rewriting them in vertical format (as they had been instructed to do) whenever they were presented horizontally? What accounts for the rich diversity of error?

These questions prompted an attempt to understand them comprehensively in a descriptive framework or extended metaphor, called "superposition of frames". The metaphor suggests that a child's mathematical knowledge is initially assimilated into fragmented, isolated frames of knowledge; when the child finds appropriate "correspondences" between frames, and brings about a "superposition of frames", then what initially had been knowledge in disarray becomes integrated into a more or less coherent body of procedural and conceptual knowledge.

In this section a perspective will be developed that will serve as a basis for this attempt to understand the study's findings, particularly children's errors and misconceptions, many examples of which are reported in Chapter 4.

The data suggest that children's developing knowledge of mathematics may be characterized at least in part as:

1. Incomplete
2. Fragmented
3. Unstable
4. Entrenched or "Set"

1. Children's developing knowledge is incomplete.

This is not to belabor the obvious but to emphasize that the exposition of any relatively complex topic extends over time. The details of the topic and all its ramifications cannot be fully presented to the children at one time or even over many times. From a constructivist perspective, the child is said to assimilate incoming information to existing cognitive structures. This is a selective process in which some of the information is retained, some not apprehended and rejected, and some simply not perceived. There are inevitably missing pieces. For example, in this study, many children could not read 3-digit numbers at pretest, and few knew the place value decimal structure of the numbering system.

2. <u>Children's knowledge is fragmented.</u>

This fragmentation is related to and is the counterpart to incomplete knowledge. Many of the missing pieces are those that if reviewed and assimilated might possibly complete what is retained into a coherent whole. Information is retained in bits and pieces that tend to be inappropriately applied or combined, especially when a child is trying to deal with new or unfamiliar material.

Example: Trying to read a 3-digit number, 776:

"One hundred seventy, seventy six"

Example: Trying to read 308:

"Three eighty ... three hundred, thirty eight"

Example: Adding single digits regardless of their place value may be viewed as an isolated piece of knowledge when applied to the problem (96+7) summed as a sequence of single digits: "9+6+7 -> 22".

Example: Another child solves the same problem (96+7) by counting seven on from 96. The correct answer is spoken aloud: "One hundred and three" but written as "1300". The one hundred and the three are unintegrated pieces clearly embedded in the answer. Another common rendition of this number is "1003".

Example: The problem (54+62) is presented in horizontal format and a subject re-writes the problem in vertical format, solving it as follows:

$$\begin{array}{r} 54 \\ \underline{+62} \\ 11 \end{array}$$

Here we can see a number of isolated pieces of knowledge being applied (the "pieces" are bracketed below):

a. [Align the symbols] --- but the plus symbol (+) participates in the alignment and displaces the tens and units of the addend.

b. [Add up each column] --- the left column is added first. A correct solution is still possible ...

c. [Stop when the tens and units places in the answer have been filled] --- in this case both digits of the sum (5+6 = 11) are put into the answer, stopping further processing.

Support for the view that children's knowledge is fragmented is found in the literature. DiSessa (1983) has noted among novice physics students a similar phenomenon which he calls "knowledge in pieces": "... intuitive physics consists of a rather large number of fragments rather than ... integrated structures ..." Young and O'Shea (1981) developed a computer simulation of children's written subtraction as a production system (a system of if-then rules) and contrast it to the view that a skill is a hierarchy of subskills. They said: "The production system analysis sees the skill as a more anarchic structure, made up from a collection of independent pieces, each representing a chunk of codified knowledge."

3. <u>Children's developing knowledge is unstable.</u>

The assimilated bits and pieces of knowledge become loosely connected into unstable, shifting, trial-and-error, idiosyncratic configurations. The following example of knowledge instability is a set of three multicolumn addition problems done by the (same!) child during one session:

Example: 13+46 -> 23

The 46 is broken into digits: (13+4+6 -> 23)

Example: 54+62 -> 96

The 54 is broken into digits (5+4 -> 9); the 6 of the 62 is appended to the 9, and the 2 is ignored.

Example: 84+67 -> 10000

The child decides the sum is some indefinitely large number.

Here is another set of problems solved by another child during one session:

Example: 26+38 -> 55   The problem was rewritten and solved as:

```
  1
 26
+38
---
 54
  5
```

Addition began in the standard way, adding the units column, entering 4 into the units answer place, then putting the carry back into the units column, recomputing the units column, and changing the 4 to 5. Finally the tens column was summed.

Example: 54+62 -> 16   Subject rewrote and solved this as:

```
 54
+62
---
 16
```

The carry from the tens was ignored or incorporated into the one in the answer. (Did the subject think of this as a "no-carry" problem?)

Example: 84+67 -> 1411   Subject rewrote this as:

```
  84
 +67
----
1411
```

This kind of instability, in which similar problems in the same paper are solved in different ways, are reported by most observers of children's errors (Brown & Burton, 1978; Brown & VanLehn, 1980).

4. <u>Children's knowledge becomes entrenched or "set"</u>

In contrast to the instability described above, children's knowledge also becomes entrenched into one of several alternative modes or approaches available to the child. This is what is described in the literature as "set effect":

> ... problem solvers become biased by their experiences to prefer certain problem solving operators. (Anderson, 1987)
>
> ... mental walls which block the problem solver from correctly perceiving a problem or conceiving its solution. (Adams, 1984)
>
> ... problem solving set --- a tendency to repeat a solution process that has been previously successful. (Glass & Holyoak, 1986)

I shall use the term "set" in a broad sense to refer not only to a general tendency to persist in some mode of operation but also to specific buggy procedures.

Here is an example of set in the broader sense. All two-digit problems in pretest and posttest were presented in horizontal format. Instruction during treatment was explicit that the problems be rewritten in vertical format. Then just as the posttest was about to begin, each subject was instructed, "Do these work sheet problems any way you want to. You can do them the way you did them on the computer, putting the numbers up and down. Or you can do them the way you learned to do them in your classroom. Any way you want to." Nevertheless, some children persisted in trying to solve the problems in horizontal format, which is difficult when carries are involved, and they often reverted to making the same kind of errors they had made in pretest. After the subject had completed all the posttests, the examiner returned to the problems done incorrectly in horizontal format and requested, "Try to do these again. Write them up and down and do them just the way you did them on the computer." Eight children were prompted this way and responded by rewriting the problem vertically. Some were able to obtain correct answers without intervention. One child even reproduced the 0-9 digit array that had been part of the screen display. Another when asked, "Why didn't you write them down 'up and down' just as you had done on the computer?", replied, "Because they were [given] this way" (gesturing horizontally across the paper). The children were "set" into solving problems in horizontal format when problems were presented in that format.

The chaotic state of affairs depicted in this section makes one wonder how learning some coherent body of mathematical knowledge is at all possible. It does happen, however, but for many children, laboriously. By the end of the third grade the great majority have mastered the multicolumn addition algorithm, although place value understanding still eludes many.

5.3 Superposition of Frames

Having developed a perspective on children's errors, the following quotation sets the stage further for an approach towards understanding this disarray of mathematical knowledge. Resnick & Ford (1981) have stated a basic dilemma of mathematics education which is expressed today as a distinction between procedural and conceptual knowledge:

> [Brownell said that] without meaningful instruction to point out the interrelationships, drill would encourage students to view mathematics as a mass of unrelated items and independent facts. ... To Thorndike, math learning consisted of a collection of bonds; to Brownell, it was an integrated set of principles and patterns. The two definitions in turn seemed to call for very different methods of teaching, either drill or meaningful instruction. Today most educators acknowledge the need for both types of learning experiences, but how they should be integrated is still not clear. (Resnick & Ford, 1981, p.19) (emphasis mine)

I shall use a metaphor, "superposition of frames" as a descriptive framework for addressing these issues and the questions raised by the study's findings. The metaphor draws heavily on the cognitive concepts of Piaget: assimilation, accommodation, equilibration (Piaget & Inhelder, 1969); and of Anderson (1983): compilation processes in production systems; and on those of the many cognitive psychologists in memory research (see the review by Baddeley, 1986). It also draws on the work of the many mathematics educators who are working in a constructivist tradition, particularly those exploring methods linking procedural and conceptual knowledge (Carpenter & Moser, 1982) and those trying to help children understand abstractions by mapping concrete experiences onto abstract symbols (Resnick & Omanson, 1987; Kamii, 1985; Fuson, 1986). It makes no claim to be a theory of cognitive processes. It is a descriptive and interpretive framework, a heuristic metaphor -- a metaphor whose terms and concepts are drawn from cognitive psychology, and a heuristic suggesting the ways information is processed by children and suggesting instructional possibilities.

This superposition-of-frames metaphor takes its departure from and is grounded in the characterizations of children's errors outlined in the previous section. A frame, as defined here, is a frame (or schema) not only in the usual large sense, "a large complex unit of knowledge that encodes typical properties of instances of general categories" (Minsky, 1975; Anderson, 1985), that is, a coherent body of knowledge, but also in the small sense, a mere isolated fragment of knowledge, as little as some obscure remembered detail. I have chosen the term "frame" rather than "schema" since it connotes boundaries and separation of knowledge, delineating a content of elements and/or procedures and/or relationships. It is <u>as if</u> this image captures the initially incomplete, fragmented character of children's knowledge.

Frames thus range in scope from the trivial to the global. We may imagine a basic attribute to be a tendency to remain as either isolated, separate modules of knowledge or clustered into associative chains --- unless a second attribute is brought into play. The second attribute is the presence of "correspondences". A frame encloses elements and procedures, of which one or more correspond to (can be mapped on to) other elements and procedures enclosed by some other frame. If a learner matches up a "correspondence" between two frames, then the two frames merge into a single composite frame, a "superposition of frames". A more (or less) coherent but integrated module of knowledge results.

What are these "correspondences"? They range from the features of a pair of analogs that are identified vaguely as "the same" to mathematical expressions identified to be precisely equivalent. Ultimately they rest on intuition: "'This' is the 'Same' as 'That'". For example, correspondences and their manner of correspondence may be seen between the following pairs of frames:

Spatial:

[An addition problem formatted horizontally]

"is the same as"

[An addition problem formatted vertically]

Analogous:

[Manipulations with physical base-ten blocks]

"is the same as"

[Manipulations with numeric symbols]

Logical:

[10 + 7 -> 17]

"is the same as"

[10 + 1 + 1 + 1 + 1 + 1 + 1 + 1 -> 17]

We have seen children for whom these pairs of frames are <u>not</u> in correspondence but remain as isolated and set frames of knowledge.

So far described, the superposition-of-frames metaphor captures the relatively incomplete, fragmented, and set character of children's errors -- but what of unstable errors? When frames are superposed, we may imagine the frames merged into a single frame containing conflicting, incompatible elements which displace one another at different times. For example, assume a child has just merged the two frames:

[Addition problem formatted horizontally]

"is the same as"

[Addition problem formatted vertically].

If the child is being taught the standard algorithm which requires right-to-left column addition, this brings into one frame the conflicting procedures:

[in horizontal format, add left-to-right]

versus

[in vertical format, add right to left].

Thus we see children adding columns in either direction, changeably; the error is unstable. If a child settles firmly on one choice, [add left-to-right], then the error becomes "set". Right-to-left processing also conflicts with standard reading patterns as well.

The educational task then is to help children find correspondences between their frames of mathematical knowledge, to help them resolve conflicts between elements within a frame, and to help them overcome "set". The next section deals with such implications of superposition-of-frames for classroom instruction.

5.4 Educational Implications

Although superposition-of-frames is but a metaphor and an application of concepts already current in psychology and education, nevertheless, it may have practical value. That is, it may suggest likely outcomes of instruction in elementary mathematics, and it may have the potential of enhancing instruction. Kilpatrick (1985) has endorsed this sort of approach in an address about reflection and recursion as metaphors in mathematics education:

> This paper has been concerned about metaphor because in my view, all our discussion about how children learn mathematics and teachers teach mathematics ultimately rests on metaphorical constructions ... (Kilpatrick, 1985)

Sections 5.4.1, 5.4.2, 5.4.3 will discuss educational implications for each of the three major phases of instruction, respectively: presentation of new material, review of material, and remediation. Section 5.4.4 discusses the general lack of understanding of place value in the sample of 36 children. Section 5.4.5 lists difficulties associated with manipulatives and suggests an alternative form of manipulative other than the standard base-ten blocks. Finally, Section 5.4.6 continues the procedural/conceptual debate and attempts a resolution.

5.4.1 Presentation of New Material

The metaphor suggests that presentation of new material, whether in the form of chalkboard exposition, graphic demonstrations, concrete models or manipulatives, and regardless of its importance or the care with which it is prepared and presented, becomes fragmented knowledge. Statements of principles and relationships may have no higher priority in the young learner's mind over even superficial details --- all are being incompletely assimilated into isolated frames in pieces or in associated chains. Behaviorist programs, with their skills hierarchies and drill and practice, and the constructivist programs with their indirect, activity-oriented, discovery approaches, both run up against this phenomenon. In short, the new material as it is being experienced and retained by the learner will be in disarray: incomplete, fragmented, unstable, and set.

5.4.2 Review of Material

Review of the material, whether in the form of drill-and-practice worksheets or in retelling, tends to suffer the same fate as the new material itself. The difference lies in a renewed opportunity to fill in missing pieces and redress the disarray of retained information. Unfortunately, this progresses haphazardly. If they have not yet decided to abandon the effort, children are generally trying to "make sense" out of the material, at least when they are attending, making connections (superpositioning frames) on their own spontaneously or under guidance, but some frames of knowledge may become more firmly set and remain isolated; others may merge inappropriately and harbor bugs in the making.

Workbooks especially contribute to this malaise. The workbook (Eicholz et al., 1985) used by the subjects in their classrooms is probably typical of its kind. Its many exercises are presented in carefully graded steps, embellished with appealing graphics, quality printing, and story situations. Although they provide very necessary practice, workbooks bear at least one serious liability. Each page presents a single type of problem and is likely to be framed by the child as an isolated experience. Once she figures out or decides on or invents or is told the answer to the first question or two, she will fill in the rest of the blanks on a page in a patterned manner. Drill and practice of a single type of problem promotes development of certain desired automatic skills but leaves knowledge fragmented and induces "set".

The Eicholtz workbook consisted almost entirely of "fill-in-the-blank"-type problems. Only a dozen pages out of a total of 336 pages called on the child to fill in more than one blank per problem. Only one page was devoted to practicing rewriting addition problems presented horizontally into vertical format ("Copy and add."). Almost invariably, problems were presented in vertical format. One consequence of this was the many instances of misaligned digits when the children wrote out whole problems in pretest/posttest. Place value exercises were presented as diagrams of bundled and loose sticks, with instructions, "Count the sticks. Ring ten.

Write the numbers" and "Trade 1 ten for 10 ones. Write the numbers."

If workbooks are to continue in the classroom -- some would abolish them (Kamii, 1985) -- worksheets should include samplings of older as well as newer material on the same page. This would counter tendencies towards "set" and would give the teacher an occasion to help children achieve a desired superposition of frames. It also helps reveal, for diagnostic and remediation purposes, frame instabilities, fragmentations, and missing pieces. For example, a review page that contained multicolumn addition problems along with related questions about place value increases the possibility that they will be perceived as relevant to each other and not isolated pieces of mathematics.

Review, in the form of repeated exposition and drill-and-practice (including worksheets), has important benefits filling in missing pieces of knowledge and automating certain desired skills, but it also bears the liabilities of entrenching buggy procedures and other knowledge disarray (unless closely monitored); of promoting a distaste for mathematics as an elaborate exercise in recall; and of impoverishing a capacity for reasoning and problem solving.

5.4.3 Remediation

When review does not suffice to redress a child's fragmented, unstable, buggy knowledge, we turn to remediation. If the remediation takes the form of a more vigorous review (extensive drill and practice) or an elaboration of detail, we may be contributing to more of the same disarray and raising an anxious and resistant defensiveness. Instead, if we were to pose questions that challenged the child's intuitions about what is true and what is not true, we might induce "frame conflict", or in Piagetian terms, disequilibrium, and bring about a repositioning of frames into desired configurations to achieve correct solutions to problems. The following is a detailed example of this approach to remediation:

Imagine a child with a carry-in-answer bug, such as:

```
  19
 +23
 312
```

The child is then presented with two problems and solves them as follows:

Problem A	Problem B
19	20
+ 4	+ 3
113	23

To each problem we imagine, metaphorically, the child bringing into working memory a frame of knowledge which is applied to each problem. The frame and its elements are imagined to be:

[<Add up a column> <place sum in answer> <right-to-left>] The element <place sum in answer> allows both 1-digit and 2-digit sums to be placed as a partial answer. Now the instructor, intending to induce "frame conflict" or disequilibrium, has the child retrieve a different but relevant frame:

> The instructor, covering up Problem A, asks, "How much is nineteen plus four?" The child responds (probably by counting up): "Twenty three". The instructor: "How then do you explain your different answer here?" (uncovers Problem A with its answer, 112).

We imagine the child is aware that she has applied an algorithm-frame and a counting-frame to the same problem but may simply shrug off the different answers. The instructor, by implying or simply stating that the answers must be the same -- this is also another frame -- is inducing the child to superpose isolated frames into one frame (the knowledge to solve a multicolumn addition problem) with conflicting elements. The conflicting elements in the frames-to-be-merged are addition-by-(buggy)algorithm and addition-by-counting. If the child becomes aware of the conflict, that the two elements lead to but then <u>must not</u> lead to different results, she may become uncertain about which element should be applied to the problem. She may then become amenable to a resolution of the conflict. Resolving the conflict should take the form of simple numeric reasoning and an appeal to intuition of what is true/not true, as in the following suggested comparison offered to the child who is also asked to discuss it:

<u>Compare Problems A and B</u>

```
 19   <---->  20
+ 4   <----> + 3
----   ????  ----
112   <---->  23
```

(Note that this also is a superposition of frames with conflicting elements.) The instructor asks the child to compare the two problems or, if necessary, prompts: "Twenty is one more than nineteen. Three is one less than four. What do you think? Should the two sums (answers) be one more or one less or the same or different from each other? Can you tell me why?" If the child can be brought to see clearly that the sums must be the same, then the process of dislodging the bug <u>by reasoning</u> has begun. If a second-grader has not already been challenged to reason about simple number relations, this may be too subtle or too complex. If such reasoning (and especially verbalizing in reasoning) is not started as early as the child enters school, we are excluding an essential aspect of learning mathematics. For example, "Two plus three is six. True or not true? ... Why do you think that?" Or, "288 is greater than 522. True or not true? ... Why do you think that?"

A major mode of instruction in elementary mathematics should be efforts to challenge logic intuitions and induce "frame conflict" (disequilibrium) by means of true/not-true-and-explain games described above. Some of the time now spent on drill-and-practice should be spent on this reasoning form of review.

There is a need to challenge children's logic intuitions in order to cultivate in children a sense of true/not-true, a sensitivity to the analytic, syntactic aspects of both language and mathematics, a sensitivity to what situations are contradictory and ambiguous. This is essential for an integration of procedural and conceptual knowledge.

5.4.4 Place Value

The 36 second-graders in this study were ill-prepared to understand the workings of multicolumn addition. Their combined score on place value understanding at pretest was only 30%, increasing modestly to 64% at posttest. Many simply recalled phrases they had heard during instructional treatment. Even those who scored well at posttest (5 out of the 10 subjects who scored 80% or better) showed signs of not understanding place value. For example, several of them misaligned digits (5+68 -> 118), and one, quoted earlier said: "You can put the 'one' (the carry) anywhere".

Scores on place value tests such as those in the pretest/posttests in this study are only crude measures of understanding. Children will parrot back phrases and explanations heard, but without understanding. More probing questions are needed. Does a child understand place value who can answer correctly the question, "How many tens in 658?" by recalling a formula that the columns are labeled "units, tens, hundreds" -- but who cannot answer the question, "What three numbers add up easily to <u>six hundred</u> [pause] <u>fifty</u> [pause] <u>eight</u>?"

In a similar finding, Cauley (1988) in a study of borrowing in subtraction in procedurally proficient children found that they have a poor grasp of place value conventions. She suggested that an understanding of the addition composition of number is necessary to fully understand place value and borrowing.

A typical adult's exposition of place value is not likely to be understood by anyone who does not already understand place value. For example, "... seven tens plus eight tens add up to fifteen tens. Fifteen tens are made up of ten tens and five tens. We place the 'five' of the fifteen in the answer place -- this 'five' stands for five tens or fifty. The ten tens remaining from the fifteen is another name for one hundred. So we place a 'one' up here into the hundreds column. This 'one' stands for ten tens or one one-hundred ... etc."

We should not be surprised when teachers opt for a simpler symbol manipulating mode: "... seven plus eight equals fifteen. Put the 'five' here and carry the 'one' up here ... etc."

Another finding in this study bearing on place value is that only 7 out of the 36 second-graders (19%) at pretest (midyear) were able to count by hundreds. When prompted "100, 200 ..." they did so easily, continuing on by themselves, but in a way analogous to "one apple, two apples, etc.", an indication that they are more likely to see the decimal structure of the number system as a verbal pattern and not as quantitative groupings.

5.4.5 Alternative Algorithms

This gap between understanding of place value and algorithmic skill in the early grades widens as other arithmetic algorithms are learned. Kamii (1985) recommends putting off the standard algorithms for addition and subtraction to the third grade; she says they should be replaced in the second grade with less efficient algorithms that make the place value aspects of multicolumn arithmetic more explicit.

The following are examples of such alternative algorithms that do make place value more explicit:

1. Left-to-right column addition with partial sums.

```
  75
 +48
 ---
 110
+ 13
 ---
 123
```

Kamii (1985) found second-graders able to do this verbally proceeding naturally from the left, adding tens first: "Seventy plus forty is one hundred ten ... etc." She also claims that the school algorithm, adding columns right-to-left, conflicts with the developing understanding of number as a "hierarchical inclusion of a system of ones within a system of tens". Also from a practical standpoint, it is important that children see that the highest place is correct.

2. Decomposition of addends. Variant of (1) above.

```
 75 ---> 70 + 5
+48 ---> 40 + 8
        110 +13 --> (100+10)+(10+3) --> 100+(10+10)+3 -->
                                        100+20+3 ---> 123
```

3. Single digit column addition in parallel.

Modelled by Peelle (1980), this algorithm is a natural extension of single column addition and it explains what is labelled an error in the standard algorithm, the "carry-in-answer" error.

```
 75            7| 5
+48   --->   + 4| 8
             ---+--
              11|13
          (11+1)| 3
              12| 3
         1 |   2| 3
              123
```

5.4.6 Manipulatives

Several researchers (Kamii, 1985; Fuson, 1986; Resnick & Omanson, 1987) believe understanding of place value is best achieved through the use of manipulatives. I have followed their lead in this study for the simulation group, where operations on base-ten blocks are closely mapped on to or correspond to operations with symbols, and vice versa. However, there are some cautions and controversies over the use of manipulatives.

The display of blocks on the computer screen was intended to substitute for physical manipulatives -- concrete embodiments of their abstract, symbolic counterparts (numbers). The manipulations of the screen blocks matched the symbol manipulations of the algorithm. Of course, the screen blocks were not "concrete", could not be physically handled, but were themselves abstract symbols. This "mapping" instruction did not prove to be successful with the children, most of whom did not see the point of it. A few complained the display was a bother. Their attention was focused almost entirely on the novelty of moving symbols around the screen with the light-pen.

Manipulatives and other concrete representations continue to appeal to educators as an effective way to bring "meaning" into mathematics. The frames metaphor would suggest this would be a complex undertaking for children; although manipulatives offer a possibility of conveying meaning, they also inject many possibilities for the kinds of knowledge disarray described above. Are the children learning the abstractions of place value? Are they learning two separate activities? Or are they learning a single complex algorithm, the blocks-symbols algorithm, without understanding place value?

Hughes (1986) discusses (and superposition-of-frames predicts) how children "translated" between concrete blocks and symbolic subtraction:

> The children observed here seemed to be only dimly aware they were dealing with two different representations of the same problem and that the two answers should agree. Rather they seemed to regard the written procedure of decomposition and the concrete manipulation of material as being two fundamentally unrelated activities. (Hughes, 1986, p.120)

Administrative problems in using manipulatives are complex. Suydam & Higgins (1978) caution that for manipulatives to be effective they should be consistent with curriculum goals, used frequently, with other aids such as diagrams and films, in the context of discovery learning, with the recording of results symbolically, and in the form of simple materials.

Jackson (1978) described a number of additudinal factors that have operated against a more widespread use of manipulatives: problems of control and management in the classroom, pressure to complete curriculum goals (where learning to manipulate symbols takes precedence over understanding), inertial resistance towards giving up worksheet assignments, attitudes that manipulatives are "kid stuff", and reaction to an overzealous acceptance of manipulatives as an educational panacea.

A more fundamental difficulty than such administrative or additudinal difficulties in modelling symbol systems is that there are always some incompatible or irrelevant features between analogs. In particular, the base-ten blocks or bundles of objects used to represent decade structure differ from multidigit numbers in fundamental ways, perceptually and structurally.

1. The relation between blocks and numbers is indirect. Groupings of blocks can represent the same total quantity as a multidigit number without corresponding to the digits of that number. For example, in base-ten blocks, 12 hundreds plus 13 tens plus 14 units represent the total quantity 1344 but not the digits of that number.

2. Physical placement is irrelevant to the total quantity represented by blocks. Blocks can be positioned in any arrangement without changing their total value, but not digits.

3. Zero and negative quantities are easily represented symbolically, but not physically. Such numbers appeared late in human history because physical

things are either physically present or not present, and only natural positive numbers were permitted. For example, Roman numerals have no symbol for zero and no place value. The numerals themselves are tick marks corresponding to physical objects or groups of other numerals (e.g., X <--> VV <--> IIIIIIIIII). There is no physical way of representing the absence of an object, although we do it today symbolically with zero.

All of these differences have to be rationalized before children are convinced (rather than coerced) that the blocks system "is the same as" the number system. The bundles of sticks or base-ten blocks are a much closer analog to the Roman numeral system, which was abandoned long ago, and the leap to multidigit numbers may be too great to be made in one step by many children. Here I suggest two intermediate steps or stages to ease the transition. All four stages are described below:

1. Counting loose objects or blocks, bundling them in groups of tens and hundreds. These are the standard base-ten manipulatives with the characteristics described above. For many children counting is solely a sequential naming process and not yet a hierarchical system of ones nested in tens.

2. Play money consisting of coin-like chips, all of the same size and diameter but differing only in color and value: Green chips have a value of 100; silver

chips, a value of 10; and copper-colored chips, each having a value of 1. The play money is used to "buy" or change for the loose and bundled objects. It has essentially the same characteristics as those objects, except that here children learn that <u>single symbols can represent groups of objects</u>. The children who have not yet mastered counting out money would also benefit by this game. Real money or simulated money should not be used and would not be appropriate for the next stage since real money values are signalled not only by color but also by size and material.

3. A new rule is applied to the coin-like chips of play money: only the least number of coins may be used to buy the objects. This is managed by using place-value trays (Figure 5.2), each capable of holding only 9 coins of one color in the following arrangement:

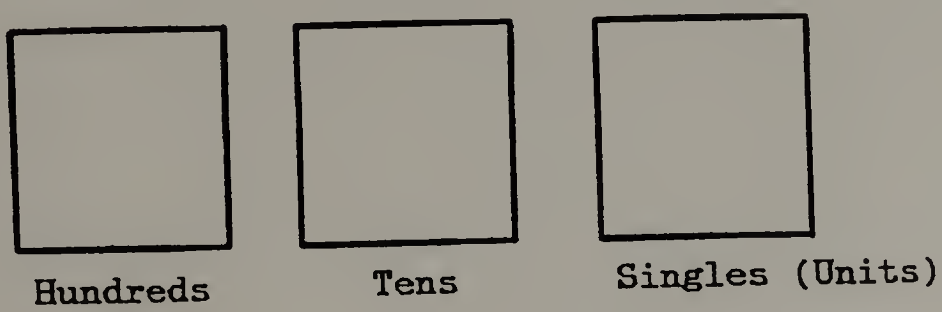

Figure 5.2 Place value trays

This arrangement emulates the counting boxes, trays, or abaci in use in medieval times prior to the introduction of the hindu-arabic numeration with zero as a new symbol.

The zero represents the absence of coins in the tray. The tray acts as a place value holder.

4. The count of the coins in each tray can now be written as a digit under each tray. The meaning of the digits may now be seen as a total value determined by the sum of the number of hundreds, tens, and singles in their respective trays. This is a much closer analog to multidigit numbers than bundles or blocks. As a further next step the trays may also be used as a quasi-abacus, as a near analog emulating the operations of multicolumn addition and subtraction. Processing might start left-to-right, as Kamii recommended above, with back-trading. Once the basic idea is learned, then the right-to-left algorithm can be introduced as minimizing back-trading.

Note the transitions from one step to another are a superposition of frames with few but distinct correspondences facilitating the transitions.

5.4.7 Procedural vs Conceptual

The focus of this study has been on algorithmic learning and would seem to put it on one side of the debate over the relative emphasis placed on procedural versus conceptual knowledge in mathematics education. However, there is an apparent paradox here; in making a procedural/conceptual distinction we may be creating a spurious dichotomy. Thus on closer examination, conceptual knowledge can be made to merge into procedural knowledge, as the following example shows. Having a conceptual knowledge of the addition algorithm implies an understanding of both the place value structure of the decimal numbering system and the structure of arithmetic (in particular, its associative and commutative laws). The meaning of these structures is itself "algorithmic", that is, syntactic, expressed in terms of elements, operators, and rules of combination -- like an algorithm. An "understanding" or proof of some mathematical relationship always emerges in the end as an exercise in syntax, sounding very much like an algorithm. Any given procedure has as its logical underpinning other procedures from which the given procedure is derived. Underpinning those are still others, until we encounter the axioms of the entire system, at which point our "understanding" stops. The axioms are "rote", accepted by convention or by intuition. Then one could argue that what children are lacking is not "conceptual" knowledge of place value but procedural knowledge of the workings of place value and of commutativity and associativity. The set of transformations (56+78) <-> (50+6+70+8) <-> (50+70+6+8) <-> (120+14) <-> (120+10+4) <-> (130+4) <-> (134) is an algorithm composed of algorithms. Where are "understanding" and conceptual knowledge now? However, this does not resolve the debate. Mathematics is not solely a set of procedures. One does not become an expert in electronics, for example, only by building circuits from kits, following directions step-by-step.

However, the debate continues with a shift in terminology: domain-specific knowledge vs. heuristics.

> Much of the emphasis on problem solving and heuristics comes from observations that students who have learned a new principle are often unable to use it intelligently to solve problems. The assumption is made that they lack suitable general problem solving strategies ... However, this failure could be explained by a lack of suitable schemas or rule automation ... Most available evidence suggests that superior problem solving skill does not derive from superior heuristics but from domain-specific skill. (Owen & Sweller, 1989)

The debate becomes altogether muddled if we ask if executing a procedure can be entirely concept-free, or if we consider where the notion of domain-specific heuristics fits in. When does a heuristic stop being a domain-specific detail and become a general heuristic or a problem solving strategy?

We are all groping towards a resolution of this debate. In elementary mathematics there is much that is rote and procedure-driven to be learned: the decimal number system itself, notation and its often inconsistent and ambiguous conventions (Iverson, 1972), the descriptive vocabulary needed to talk about mathematics, and all the algorithms in arithmetic that provide children with practical tools. Behaviorists, however, underestimate the extent to which emphasis on skills acquisition and drill and practice promotes "set", entrenches bugs, and fosters a preoccupation with memorized sequences. Constructivists, emphasizing discovery methods and the unifying concepts that impart power and beauty to abstract structures, underestimate the need to automate skills, the time and effort needed to diagnose and remediate knowledge in disarray, and the extent to which children discover and embrace buggy methods unless closely monitored.

The superposition-of-frames metaphor being proposed here, with its description of likely outcomes of instruction and the tactics it suggests for instruction, may provide a means of integrating these two kinds of instruction.

5.5 Responses to General Questions

We are now in a better position to respond to the general questions raised in the introductory chapter:

> May understanding and the ability to manipulate symbols proceed independently of one another, at least for a short time?

Yes. Initially both are likely to be framed as independent frames of knowledge, until the learner matches up correspondences and brings about a superposition of frames and an integration of both. The significantly better performance of the assisted group in this study suggests that separation initially is also necessary to avoid overloading processing capacity.

> To what extent does having an understanding of mathematical principles facilitate learning an algorithm? Conversely, does the learning of an algorithm facilitate understanding mathematical principles?

Algorithms/principles are terms equivalent to the terms procedural/conceptual discussed in Section 5.4.7 above. That discussion applies here as well. Algorithms/principles can be mutually facilitating -- when algorithms may be seen as instances of applied principles, and conversely, when principles may be seen as (abstracted from) correspondences between instances of algorithms.

> What aspects of an algorithm tend to emerge as buggy procedures?

When the errors of the children in this study are examined, it seems that hardly any aspect of the algorithm escapes being converted into some kind of bug.

5.6 Suggestions for Further Research

The methods, findings, and interpretations of data in this study suggest further research in the area of learning algorithms and acquiring place value understanding. The following are suggestions in the form of research questions:

1. Variants of the current study.

   a) Would stronger effects emerge if the number of instructional sessions were increased?

   b) If a computer addition problem were alternated with an equivalent pencil-on-paper problem, would this closer juxtaposition of symbol systems facilitate transfer between the two media and result in more effective learning of the algorithm?

   c) Would redesigning the screen displays improve learning? For example, the placement of the "memory box", or placing the tableau of numbers on the right side of the screen. The arrangement and sequencing of displays are not necessarily optimum.

   d) Would allowing the learner to interact with the block displays with the light pen, instead being a passive viewer, improve learning? Allow the learner to manipulate the simulated blocks with the light pen.

2. Children's notion of zero. When do children go beyond "Zero is nothing" and understand its role as a place value holder? Pose a number of questions about

numbers containing zero, of children spanning grades 2 to 6.

3. Place value.

   a) Would teaching the less efficient but more explicit algorithms described in Section 5.4.4 (Place value) enhance understanding of place value?

   b) Would the transitional stages in manipulatives involving uniformly sized play money and the place value trays described in Section 5.4.5 (Manipulatives) facilitate understanding of place value?

4. The superposition-of-frames metaphor. Does the superposition-of-frames metaphor have any value in classroom instruction? Develop a sequence of questions that induce "frame conflict" which is subsequently resolved by reasoning quantitatively. (See Section 5.4.3)

5. Alternative addition algorithms. Would the learning of alternative addition algorithms described in Section 5.4.5 enhance place value understanding?

6. Subtraction. Would the methods used in this study of the addition algorithm apply to a study of the standard school algorithm for subtraction?

5.7 Summary and Conclusions

1. Evidence has been found that learning multicolumn addition by means of software developed for this study is more effective when on-screen number-fact assistance is provided. The general conclusion is that an algorithm is learned more effectively if some of the demand on short term memory is lifted temporarily, such as the child's effort to recall or reconstruct number-facts or the instructor's imposition of explanatory material.

2. The version of the software designed to enhance place value understanding by simultaneously displaying simulated blocks which represent the symbol manipulations of the algorithm, was found to be only partially effective. This finding is consistent with although weaker than (1) above. The simulation displays and instructor's explanations of place value were possibly an additional load on the child's limited processing capacity. Consequently, significantly higher scores occurred in place value understanding but not in algorithmic performance.

3. A metaphor has been proposed to account for anomalies in the findings and to understand the rich diversity of errors displayed by the children in multidigit addition. The metaphor, called "superposition of frames", suggests that children's mathematical

knowledge is fragmented into isolated frames of knowledge. When a child finds appropriate "correspondences" between frames, and brings about a "superposition of frames", what had been knowledge in disarray becomes integrated into a coherent body of procedural and conceptual knowledge. The metaphor may have value in providing a parsimonious description of the likely outcomes of instruction and in suggesting instructional tactics for helping children to integrate their mathematical knowledge.

4. Multicolumn addition and subtraction provide rich opportunities for educational research. A number of suggestions were made in this study for further research, particularly in the use of computers and concrete manipulatives in learning algorithms and understanding place value.

APPENDICES

APPENDIX A

FORMATION OF CONTROL AND TREATMENT GROUPS

To form three equalized groups which would be randomly assigned as control or one of the two treatment groups, subjects were first scored in the pretest with one point for each problem or question correctly answered.

A composite score was then derived by assigning 1 point to categories of the pretests for correctly answering a minimum number of problems in the category, as set forth in Table A.1 below:

TABLE A.1 Pretest Composite Score

	Min Score to obtain 1 Point
A. Basic Skills	
1. Counting	4 correct out of 6
2. One-digit addition	11 correct out of 15
3. Read 2-, 3-digit numbers	3 correct out of 3
4. Counting money	3 correct out of 4
B. Place Value	
5. What does the digit mean?	4 correct out of 8
6. Which number is larger?	2 correct out of 2
7. How many tens/hundreds?	3 correct out of 3
8. Name tens	2 correct out of 2
9. Same digit in diff. pos.	1 correct out of 1
10. Decomposition	1 correct out of 1
11. Composition	1 correct out of 1
12. Number proximity (1-digit)	1 correct out of 1
13. Number proximity (3-digit)	1 correct out of 1
C. Multicolumn Addition	
14. to 21. Eight problems, one point each	
Maximum composite score	21

The composite score for each subject was then ranked. Three groups were formed by an "equalizing method" by assigning high and low rankings to each group. Sex and classroom were also equalized. The intended result was to form three statistically comparable groups which were randomly assigned as control, assisted, and simulation groups. The random assignment was performed by writing the names of the groups on separate slips of paper, mixing up the slips, and assigning each equalized group to each slip as it was picked in turn.

To confirm the validity of this particular method of weighting the pretest scores, alternative weighting methods were applied to the three groups and compared statistically by means of a one-way analysis of variance. No significant differences were found among any of the five scoring methods. See Table A.2 on the following page.

TABLE A.2 Evaluation of Pretest Weighting Methods

Group:	Control	Assisted	Simulation	p-value
Method 1 (Maximum score: 22)				.932
Mean	8.7	9.2	8.8	
SD	4.7	2.3	3.1	
Method 2 (Maximum score: 57)				.913
Mean	30.7	32.0	31.7	
SD	10.0	6.2	7.2	
Method 3 (Maximum score: 100%)				.966
Mean	47.0%	48.4%	47.2%	
SD	16.9	11.4	12.6	
Method 4 (Maximum score: 25)				.988
Mean	10.0	10.2	9.9	
SD	4.6	3.2	3.3	
Method 5 (Maximum score: 100%)				.944
Mean	45.8%	47.7%	46.5%	
SD	17.5	11.6	13.2	

Method 1: The original weighting method described above.

Method 2: One point for all problems.

Method 3: Composite percentage. Based on one point for each problem, each category (basic skills, place value, multicolumn addition) given a percent score, and then these three percentages were averaged.

Method 4: Subcategory scoring. Each problem within a subcategory in the basic skills and place value categories was given fractional scores, in effect, giving each subcategory a score of one. The addition problems were each given a score of one. Then the total was obtained.

Method 5: Composite percentage. Based on giving a percent score to each category in Method 4 and combining these three equally into a single percentage.

APPENDIX B

DATA TABLES

TABLE B.1 Subject Scores

	MAX SCORE	CONTROL GROUP					
SUBJECT		AD	TAN	JES	LV	BC	JER
CLASSROOM		G	G	G	G	G	G
SEX		F	F	F	F	M	M
GROUP ASSIGNMENT		C	C	C	C	C	C
STARTING DAY		41	58	41	58	32	74
ENDING DAY		48	62	46	62	41	79
------ BASICS ------							
COUNTING	6	6	3	5	4	4	4
ORAL ADDITION	9	8	5	5	7	9	9
WRITTEN ADDITION	6	5	6	6	5	5	6
READ NUMBERS	3	3	1	0	2	2	2
COUNTING MONEY	4	4	0	1	4	4	4
TOTALS	28	26	15	17	22	24	25
------ PLACE VALUE ------							
PRETEST	21	15	4	4	4	5	3
POSTTEST	21	17	7	6	6	16	8
DIFFERENCE		2	3	2	2	11	5
-------- ADDITION --------							
PRETEST	8	4	2	2	2	2	3
POSTTEST	8	3	3	1	8	1	1
DIFFERENCE		-1	1	-1	6	-1	-2
MONITOR CHECK	3	3	0	0	2	0	0
TRANSFER PROBLEMS	6	3	1	0	5	1	0
CORRECTION PROBLEMS	4	2	0	0	4	3	0

Continued, next page.

Table B.1 (Continued)

		CONTROL GROUP (CONTINUED)						
SUBJECT		CC	KF	DF	GM	FR	PS	TOTAL
CLASSROOM		F	F	F	F	F	F	
SEX		F	F	M	M	M	M	
GROUP ASSIGNMENT		C	C	C	C	C	C	
STARTING DAY		69	25	39	62	62	48	
ENDING DAY		74	34	46	67	67	48	
	MAX SCORE							
------ BASICS ------								
COUNTING	6	3	6	1	5	2	5	48
ORAL ADDITION	9	8	8	4	8	9	9	89
WRITTEN ADDITION	6	6	6	6	6	6	6	69
READ NUMBERS	3	1	2	1	2	1	3	20
COUNTING MONEY	4	1	4	0	4	1	4	31
TOTALS	28	19	26	12	25	19	27	257
------ PLACE VALUE ------								
PRETEST	21	1	12	3	10	3	15	79
POSTTEST	21	3	17	8	16	12	21	137
DIFFERENCE		2	5	5	6	9	6	58
-------- ADDITION --------								
PRETEST	8	0	4	1	3	4	3	30
POSTTEST	8	5	6	2	8	4	6	48
DIFFERENCE		5	2	1	5	0	3	18
MONITOR CHECK	3	0	1	0	3	1	1	11
TRANSFER PROBLEMS	6	5	4	1	4	0	2	26
CORRECTION PROBLEMS	4	1	4	0	3	0	1	18

Continued, next page.

Table B.1 (Continued)

		ASSISTED GROUP					
SUBJECT		JL	ER	OR	MIS	NA	BED
CLASSROOM		G	G	G	G	G	G
SEX		F	F	F	F	M	M
GROUP ASSIGNMENT		A	A	A	A	A	A
STARTING DAY		65	58	74	25	25	32
ENDING DAY		69	62	79	34	34	41
	MAX SCORE						
------ BASICS ------							
COUNTING	6	4	3	4	5	6	5
ORAL ADDITION	9	9	9	8	9	7	7
WRITTEN ADDITION	6	6	6	6	6	6	4
READ NUMBERS	3	1	1	1	1	3	3
COUNTING MONEY	4	3	3	4	4	4	1
TOTALS	28	23	22	23	25	26	20
------ PLACE VALUE ------							
PRETEST	21	1	5	7	2	11	12
POSTTEST	21	9	15	16	18	21	18
DIFFERENCE		8	10	9	16	10	6
-------- ADDITION --------							
PRETEST	8	1	4	3	2	2	4
POSTTEST	8	7	8	7	8	7	8
DIFFERENCE		6	4	4	6	5	4
MONITOR CHECK	3	0	1	1	1	0	3
TRANSFER PROBLEMS	6	3	4	6	5	5	4
CORRECTION PROBLEMS	4	3	3	3	4	2	3

Continued, next page.

Table B.1 (Continued)

		ASSISTED GROUP (CONTINUED)						
SUBJECT		JUR	JSS	CG	EP	AR	TR	TOTAL
CLASSROOM		G	G	F	F	F	F	
SEX		M	M	F	F	F	M	
GROUP ASSIGNMENT		A	A	A	A	A	A	
STARTING DAY		58	69	27	44	25	60	
ENDING DAY		62	74	34	48	34	65	
	MAX SCORE							
------ BASICS ------								
COUNTING	6	5	4	5	5	5	4	55
ORAL ADDITION	9	8	9	9	7	9	8	99
WRITTEN ADDITION	6	6	5	6	5	6	6	68
READ NUMBERS	3	3	1	2	2	3	3	24
COUNTING MONEY	4	4	3	1	0	4	4	35
TOTALS	28	26	22	23	19	27	25	281
------ PLACE VALUE ------								
PRETEST	21	9	1	1	5	12	5	71
POSTTEST	21	16	11	8	16	18	8	174
DIFFERENCE		7	10	7	11	6	3	103
-------- ADDITION --------								
PRETEST	8	4	3	2	1	4	2	32
POSTTEST	8	7	5	3	6	4	3	73
DIFFERENCE		3	2	1	5	0	1	41
MONITOR CHECK	3	0	0	0	0	0	0	6
TRANSFER PROBLEMS	6	5	4	1	0	6	6	49
CORRECTION PROBLEMS	4	3	4	0	0	4	3	32

Continued, next page.

Table B.1 (Continued)

		SIMULATION GROUP					
SUBJECT		KEF	KH	EL	XV	VCL	AL
CLASSROOM		G	G	G	G	G	G
SEX		F	F	F	F	M	M
GROUP ASSIGNMENT		B	B	B	B	B	B
STARTING DAY		46	25	65	76	65	27
ENDING DAY		60	34	72	88	72	39
	MAX SCORE						
------ BASICS ------							
COUNTING	6	4	3	4	5	3	5
ORAL ADDITION	9	8	7	6	9	8	9
WRITTEN ADDITION	6	6	6	5	6	6	6
READ NUMBERS	3	1	2	1	3	3	1
COUNTING MONEY	4	1	4	1	4	4	4
TOTALS	28	20	22	17	27	24	25
------ PLACE VALUE ------							
PRETEST	21	5	8	6	14	10	3
POSTTEST	21	14	14	11	14	19	10
DIFFERENCE		9	6	5	0	9	7
-------- ADDITION --------							
PRETEST	8	1	2	2	4	3	2
POSTTEST	8	3	7	2	7	3	8
DIFFERENCE		2	5	0	3	0	6
MONITOR CHECK	3	0	3	0	3	2	0
TRANSFER PROBLEMS	6	2	3	1	5	6	5
CORRECTION PROBLEMS	4	2	3	1	4	3	0

Continued, next page.

Table B.1 (Continued)

		SIMULATION GROUP (CONTINUED)						
SUBJECT		JOR	BG	JK	VR	DB	MIR	TOTAL
CLASSROOM		G	F	F	F	F	F	
SEX		M	F	F	F	M	M	
GROUP ASSIGNMENT		B	B	B	B	B	B	
STARTING DAY		44	25	32	74	27	67	
ENDING DAY		48	34	41	81	39	72	
	MAX SCORE							
------ BASICS ------								
COUNTING	6	3	5	3	2	6	6	49
ORAL ADDITION	9	8	8	9	6	9	9	96
WRITTEN ADDITION	6	5	6	6	6	6	6	70
READ NUMBERS	3	3	2	3	0	2	3	24
COUNTING MONEY	4	1	3	4	3	4	4	37
TOTALS	28	20	24	25	17	27	28	276
------ PLACE VALUE ------								
PRETEST	21	4	3	4	3	5	11	76
POSTTEST	21	17	15	11	14	19	18	176
DIFFERENCE		13	12	7	11	14	7	100
-------- ADDITION --------								
PRETEST	8	1	2	3	1	5	2	28
POSTTEST	8	2	3	3	2	6	6	52
DIFFERENCE		1	1	0	1	1	4	24
MONITOR CHECK	3	1	0	0	0	1	0	10
TRANSFER PROBLEMS	6	1	1	1	0	2	2	29
CORRECTION PROBLEMS	4	0	0	0	1	1	2	17

TABLE B.2 Time Data

	CONTROL GROUP					
NAME	AD	TAN	JES	LV	HC	JER
CLASSROOM	G	G	G	G	G	G
SEX	F	F	F	F	M	M
GROUP ASSIGNMENT	C	C	C	C	C	C
SCORE (MAX: 6)	4	3	2	5	6	4
SIGNIFICANT ERRORS	2	6	12	1	0	2
"THRASHING" ERRORS	4	28	36	7	3	8
TIME IN CORRECT MOVES (sec)	356	321	316	209	231	179
TOTAL TIME (sec)	380	439	483	246	248	201
PERCENT TIME IN CORRECT MOVES	94	73	65	85	93	89

Continued, next page.

Table B.2 (Continued)

	CONTROL GROUP (CONTINUED)						
NAME	CC	KF	DF	GM	FR	PS	TOTAL
CLASSROOM	F	F	F	F	F	F	
SEX	F	F	M	M	M	M	
GROUP ASSIGNMENT	C	C	C	C	C	C	
SCORE (MAX: 6)	3	6	0	4	5	3	45
SIGNIFICANT ERRORS	4	0	14	3	1	3	48
"THRASHING" ERRORS	11	10	43	13	10	15	188
TIME IN CORRECT MOVES (sec)	387	227	245	213	209	239	3132
TOTAL TIME (sec)	438	241	526	254	228	295	3979
PERCENT TIME IN CORRECT MOVES	88	94	47	84	92	81	79

Continued, next page.

Table B.2 (Continued)

	ASSISTED GROUP					
NAME	JL	ER	OR	MIS	NA	BED
CLASSROOM	G	G	G	G	G	G
SEX	F	F	F	F	M	M
GROUP ASSIGNMENT	A	A	A	A	A	A
SCORE (MAX: 6)	3	5	4	4	4	4
SIGNIFICANT ERRORS	5	1	3	2	3	2
"THRASHING" ERRORS	23	10	35	6	3	8
TIME IN CORRECT MOVES (sec)	259	216	236	172	208	248
TOTAL TIME (sec)	355	239	333	191	238	333
PERCENT TIME IN CORRECT MOVES	73	90	71	90	87	74

Continued, next page.

Table B.2 (Continued)

	ASSISTED GROUP (CONTINUED)						
NAME	JUR	JSS	CG	EP	AR	TR	TOTAL
CLASSROOM	G	G	F	F	F	F	
SEX	M	M	F	F	F	M	
GROUP ASSIGNMENT	A	A	A	A	A	A	
SCORE (MAX: 6)	4	3	0	5	5	5	46
SIGNIFICANT ERRORS	2	5	12	1	1	2	39
"THRASHING" ERRORS	8	15	17	17	7	4	153
TIME IN CORRECT MOVES (sec)	174	193	280	196	244	219	2645
TOTAL TIME (sec)	193	245	358	227	263	230	3205
PERCENT TIME IN CORRECT MOVES	90	79	78	86	93	95	83

Continued, next page.

Table B.2 (Continued)

	SIMULATION GROUP					
NAME	KEF	KH	EL	XV	VCL	AL
CLASSROOM	G	G	G	G	G	G
SEX	F	F	F	F	M	M
GROUP ASSIGNMENT	B	B	B	B	B	B
SCORE (MAX: 6)	5	4	6	3	3	5
SIGNIFICANT ERRORS	1	2	0	4	3	1
"THRASHING" ERRORS	6	3	11	27	4	8
TIME IN CORRECT MOVES (sec)	239	298	202	209	215	183
TOTAL TIME (sec)	278	336	246	269	249	195
PERCENT TIME IN CORRECT MOVES	86	89	82	78	86	94

Continued, next page.

Table B.2 (Continued)

	SIMULATION GROUP (CONTINUED)						
NAME	JOR	BG	JK	VR	DB	MIR	TOTAL
CLASSROOM	G	F	F	F	F	F	
SEX	M	F	F	F	M	M	
GROUP ASSIGNMENT	B	B	B	B	B	B	
SCORE (MAX: 6)	0	5	4	2	4	3	44
SIGNIFICANT ERRORS	15	1	2	8	3	4	44
"THRASHING" ERRORS	39	14	4	31	3	10	160
TIME IN CORRECT MOVES (sec)	337	192	208	314	182	196	2775
TOTAL TIME (sec)	621	265	222	530	201	254	3666
PERCENT TIME IN CORRECT MOVES	54	72	94	59	91	77	76

APPENDIX C

CORRELATION MATRIX

Table C.1 Correlation Matrix

	GROUP	ROOM	SEX	START	BASICS	PREPV	POSTPV	PREADD	POSTAD	TRANS	CORR
GROUP	1.00										
ROOM	-0.70	1.00									
SEX	0.07	-0.04	1.00								
START	-0.06	-0.14	-0.04	1.00							
BASICS	0.17	0.06	-0.20	-0.17	1.00						
PREPV	-0.30	-0.02	-0.13	-0.09	0.49 *	1.00					
POSTPV	0.29	0.01	-0.26	-0.30	0.52 *	0.66 *	1.00				
PREADD	-0.06	-0.02	-0.19	-0.09	0.51 *	0.53 *	0.44	1.00			
POSTAD	0.06	-0.12	-0.00	-0.05	0.39	0.27	0.30	0.27	1.00		
TRANS	0.05	-0.23	-0.05	0.06	0.47	0.34	0.19	0.26	0.60 *	1.00	
CORR	-0.02	-0.30	0.07	0.11	0.44	0.40	0.31	0.34	0.52 *	0.74 *	1.00

* Significant at the 5% level

GROUP	Group designation (control, assisted, simulation)
ROOM	Room F,G
SEX	Male, Female
START	Starting date of instruction (Julian date)
BASICS	Basics scores
PREPV	Pretest scores - place value
POSTPV	Posttest scores - place value
PREADD	Pretest scores - addition
POSTAD	Posttest scores - addition
TRANS	Transfer test scores
CORR	Correction test scores

REFERENCES

REFERENCES

Adams, J. L. (1974). Conceptual blockbusting. New York: Freeman.

Allardice, B. S., & Ginsburg, H. P. (1983). Children's psychological difficulties in mathematics. In H. P. Ginsburg (Ed.), The development of mathematical thinking. New York: Academic Press.

Anderson, J. R. (1985). Cognitive psychology and its implications (2nd ed.). New York: Freeman.

Anderson J. R. (1982). Acquisition of cognitive skill. Psychological Review, 89, 369-06.

Anderson J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard.

Anderson J. R. (1987). Skill acquisition: compilation of weak-method problem solutions. Psychological Review, 94, 192-210.

Ausubel, D. P., & Robinson, F. G. (1969). School learning: an introduction to educational psychology. New York: Holt, Reinhart, & Winston.

Baddeley, A. (1986). Working memory. Oxford: Clarendon Press.

Bartlett F. C. (1932). Remembering: a study in experimental and social psychology. Cambridge University Press.

Brownell W. A. (1935). Psychological considerations in thelearning and the teaching of arithmetic. The teaching of arithmetic: the tenth yearbook of the National Council of Teachers of Mathematics. New York: Teachers College, Columbia University.

Brown J. S., & Burton R. R. (1978). Diagnostic models forprocedural bugs in basic mathematical skills. Cognitive Science, 2, 155-192.

Brown J. S., & VanLehn K. (1980). Repair theory: a generative theory of bugs in procedural skills. Cognitive Science, 4, 379-462.

Bruner J. S. (1962). The process of education. Cambridge, MA: Harvard University Press.

Buyers V., & Erlwanger S. (1985). Memory in mathematics understanding. Educational Studies in Mathematics, 16, 259-281.

Carpenter, T. P., & Moser, J. M. (1982). The development of addition and subtraction problem solving skills. In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), Addition and subtraction: a cognitive perspective. Hillsdale, NJ: Erlbaum.

Carpenter T. P. (1985). Research on the role of structure in mathematics. Arithmetic Teacher, 33, 58-60.

Case R. (1982). General influences in the acquisition of elementary concepts and algorithms in arithmetic. In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), Addition and subtraction: a cognitive perspective. Hillsdale, NJ: Erlbaum.

Case R., Kurland D. M., & Goldberg J. (1982). Operational efficiency and the growth of short term memory span. J. of Experimental Child Psychology, 33, 386-404.

Cauley, K. M. (1988). Construction of logical knowledge: study of borrowing in subtraction. J. of Educational Psychology, 80, 202-205.

Clements, M. A. (1982). Careless errors made by sixth-grade children on written mathematical tasks. J. for Research in Mathematics Education, 13, 136-144.

Cooper G., & Sweller J. (1987). Effects of schema acquisition and rule automation on mathematics problem solving transfer. J. of Educational Psychology, 79, 347-362.

Davis, R. B. (1986). How many ways can you understand? Arithmetic Teacher, 34, 3.

Davis, R. B. (1988). The interplay of algebra, geometry, and logic. J. of Mathematical Behavior, 7, 9-28.

Davis, R. B. (1988). The interplay of algebra, geometry, and logic. J. of Mathematical Behavior, 7, 9-28.

Dickson, W. P. (1985). Thought provoking software: juxtaposing symbol systems. Educational Researcher, 14, 30-38.

Dienes, Z. P. (1963). An experimental study of mathematics learning. London: Hutchinson Educational Ltd.

DiSessa A. A. (1988). Knowledge in pieces. In G. Forman (Ed.), Constructivism in the computer age,. Hillsdale, NJ: Erlbaum.

Eicholz, R. E., O'Doffer, P. G., Fleenor, C. R., et al. (1985). Addison-Wesley Mathematics. Menlo Park, CA: Addison-Wesley.

Engelhardt, J. M. (1982). Using computational errors in diagnostic teaching. Arithmetic Teacher, 29, 16-19.

Freemont H. (1971). New mathematics and old dilemmas. in D. B. Aichele & R.E. Reyes (Eds.), Readings in secondary school mathematics. New York: Prindle, Weber, & Schmidt.

Fuson, K. C. (1982). An analysis of the counting-on solution procedure in addition. In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), Addition and subtraction: a cognitive perspective. Hillsdale, NJ: Erlbaum.

Fuson, K. C. (1986). Roles of representation and verbalization in the teaching of multicolumn addition and subtraction. European J. of Psychology of Education, 1, 35-56.

Fuson, K. C. (1989). Children's representation of multidigit numbers: implications for addition and subtraction and place value learning and teaching. (under review).

Fuson, K. C. & Briars, D. J. (1989). Base-ten blocks as a first and second grade learning/teaching setting for multidigit addition and subtraction and place value concepts. J. for Research in Mathematics Education. (in press).

Fuson, K. C., Stigler, J. W., & Bartsch, K. (1988). Grade placement of addition and subtraction topics in Japan, Mainland China, the Soviet Union, Taiwan, and the United States. J. for Research in Mathematics Education, 19, 449-456.

Gagne, R. M. (1970). The conditions of learning. New York: Holt, Reinhart, & Winston.

Gagne, R. M. (1983). Some issues in the psychology of mathematics instruction. J. for Research in Mathematics Education, 1, 7-18.

Gelman R., & Gallistel C. R. (1978). The child's understanding of number. Cambridge, MA: Harvard University Press.

Ginsburg, H. (1977). Children's Arithmetic: the learning process. New York: Van Nostrand.

Ginsburg H. P. (1980). Children's surprising knowledge of arithmetic. Arithmetic Teacher, 28, 2-44.

Ginsburg H. P., Kossan N. E., Schwartz R., & Swanson D. (1983) Protocol methods in research on mathematical thinking. in H. P. Ginsburg (Ed.), The development of mathematical thinking. New York: Academic Press.

Glass A. L., & Holyoak K. J. (1986). Cognition. New York:Random House.

Greeno J. G. (1973). The structure of memory and the process of solving problems. In R. L. Solso (Ed.), Contemporary issues in cognitive psychology: the Loyola Symposium. Winston Publishing.

Groen G. J., & Parkman J. M. (1972). A chronometric analysis of simple addition. Psychological Review, 9, 329-343.

Hiebert J. (Ed.). (1986). Conceptual and procedural knowledge: the case of mathematics. Hillsdale, NJ: Erlbaum.

Hiebert J., & Lefevre P. (1986). Conceptual and procedural knowledge in mathematics: an introductory analysis. in J. Hiebert (Ed.), Conceptual and procedural knowledge: The case of mathematics. Hillsdale, NJ: Erlbaum.

Houlihan, D. M., & Ginsburg, H. P. (1981). The addition methods of first- and second-grade children. J. for Research in Mathematics Education, 12, 95-106.

Howson, G., Keitel, C., & Kilpatrick, J. (1981). Curriculum development in mathematics. New York: Cambridge University Press.

Hughes, M. (1986). Children and number: difficulties in learning mathematics. New York: Basil Blackwell.

Inhelder B., & Piaget J. (1964). The early growth of logic in the child. New York: Harper & Row.

Jackson, R. (1979). Hands-on math: misconceptions and abuses. Learning, 7, 76-78.

Janke, R. W. & Pilkey, P. J. (1985). Microcomputer diagnosis of whole number computational errors. J. of Computers in Mathematics and Science Teaching, 5, 45-51.

McDonald, J., Beal, J., & Ayers, F. (1987). Computer administered testing: diagnosis of addition computational skills in children. J. of Computers in Mathematics and Science Teaching, 7, 38-43.

Kamii, C. K. (1985). Young children reinvent arithmetic: implications of Piaget's theory. New York: Teacher's College Press.

Kamii C. (1986). Place value: an explanation of its difficulty and educational implications for the primary grades. J. of Research in Childhood Education, 1, 75-86.

Kamii C., & Joseph L. (1988). Teaching place value and double column addition. Arithmetic Teacher, 35, 48-52.

Kilpatrick, J. (1985). Reflection and recursion. Educational Studies in Mathematics, 16, 1-26.

Larkin, J. H., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. Science, 208, 1335-1342.

Lesgold A. M. (1984). Acquiring expertise. In J. R. Anderson & S. M. Kosslyn (Eds.), Tutorials in learning and memory. San Francisco, CA: Freeman.

Madell, R. (1985). Children's natural processes. Arithmetic Teacher, 32, 20-22.

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), The psychology of computer vision. New York: McGraw-Hill.

Morris J. (1981). Math anxiety: how to avoid it. Mathematics Teacher, 4, 6.

O'Shea, T., Evertsz, R., Hennesey, S., & Et Al. (1987). Design choices for an intelligent arithmetic tutor. In J. Self (Ed.), Artificial intelligence and human learning: intelligent CAI. New York: Chapman & Hall Computing.

Owen, E., & Sweller, J. (1989). Should problem solving be used as a learning device in mathematics? J. for Research in Mathematics Education, 20, 322-328.

Papert S. (1980). Mindstorms: children, computers, and powerful ideas. NY: Basic Books.

Peelle, H. A. (1980). Alternative addition algorithms in APL: implications for education. In G. A. van der Linden (Ed.), APL 80. North-Holland Publishing.

Piaget, J., & Inhelder, B. (1969). The psychology of the child. New York: Basic Books.

Piaget J. (1965). The child's conception of number. New York: Norton. (Original work published 1941).

Reiser R. A. (1987). Instructional technology. In R. M. Gagne (Ed.), Instructional technology: Foundations. Hillsdale, NJ: Erlbaum.

Resnick, L. B. (1982). Syntax and semantics in learning to subtract. In T. P. Carpenter, J. M. Moser, & Romberg, T.A. (Eds.), Addition and subtraction: a cognitive perspective. Hillsdale, NJ: Erlbaum.

Resnick, L. B., Nesher, P., Leonard, F., & Et Al. (1989) Conceptual bases of arithmetic errors: the case of decimal numbers. J. for Research in Mathematics Education, 20, 8-27.

Resnick L. B., & Ford W. W. (1981). The psychology of mathematics for instruction. Hillsdale, NJ: Erlbaum.

Resnick L. B., & Omanson S. F. (1987). Learning to understand arithmetic. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 3). Hillsdale, NJ: Erlbaum.

Romberg T. A. (1982). An emerging paradigm for research on addition and subtraction skills. In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), Addition and subtraction: a cognitive perspective. Hillsdale, NJ: Erlbaum.

Russell R. L., & Ginsburg H. P. (1984). Cognitive analysis of children's mathematics difficulties. Cognition and Instruction, 1, 217-244.

Schoenfeld, A. H. (1985). Mathematical problem solving. Orlando, FL: Academic Press.

Selfridge O. G. (1959). Pandemonium: a paradigm for learning. In Symposium on the mechanization of thought processes. London: HM Stationery Office.

Siders, J. A., Siders, J. Z., & Wilson, R. M. (1985). A screening procedure to identify children having difficulties in arithmetic. J. for Research in Mathematics Education, 16, 356-363.

Skemp R. R. (1987). The psychology of learning mathematics (expanded American edition). Hillsdale, NJ: Erlbaum.

Sleeman, D., & Brown, J. S. (. (1982). Intelligent tutoring systems. Orlando, FL: Academic Press.

Steffe L. P. (1983). Children's algorithms as schemes. Educational Studies in Mathematics, 1, 109-125.

Steffe L. P., Von Glasersfeld E., Richards E., & Cobb P. (1983) Children's counting types. New York: Praeger.

Stein, S. K. (1988). Gresham's law: algorithm drives out thought. J. of Mathematical Behavior, 7, 79-84.

Stevenson, H. W., Lee, S., & Stigler, J. W. (1986). Mathematics achievement of Chinese, Japanese, and American children. Science, 231, 693-699.

Suydam, M. N. (1975). Algorithmic learning. In M. N. Suydam & A. R. Osborne (Eds.), Algorithmic learning, (ERIC Document Reproduction Service No. ED-113-152).

Suydam, M. N., & Higgins, J. L. (1976). Review and synthesis of studies of activity-based approaches to mathematics teaching (Final Report, NIE Contract No. 400-75-0063). Washington, DC: National Institute of Education.

Taylor R. P. (Ed.). (1980). The computer in the school: tutor, tool, tutee. New York: Teachers College Press.

Treffers A. (1987). Integrated column arithmetic according to progressive schematization. Educational Studies in Mathematics, 18, 125-145.

Wearne, D., & Hiebert, J. (1988). A cognitive approach to meaningful mathematics instruction: testing a local theory using decimal numbers. J. for Research in Mathematics Education, 19, 371-384.

Young R. M., & O'Shea T. (1981). Errors in children's subtraction. Cognitive Science, 5, 153-177.