Robotics & Computer-Integrated Manufacturing, Vol 5, No 1, pp 33–42, 1989
Printed in Great Britain

0736-5845/89$3 00 + 0 00
Pergamon Press plc

• *Paper*

CAPTURING THE DESIGN OF MECHANICAL COMPONENTS IN VLSIs*

YORAM KOREN and AVI LOWY

Department of Mechanical Engineering and Applied Mechanics, The University of Michigan, Ann Arbor, MI 48109, U.S.A.

This paper proposes an approach for implementing mechanical computer-aided design (CAD) systems with the aid of generic, multi-purpose VLSIs, as a partial substitute for software. Although software is not eliminated, a more equitable share of tasks is reached between software and hardware, providing for faster computer execution.

Two approaches are presented; the first is based on "wired-in" design rules for every permissible solution, feasible but not practical; the second an approach which is based on a "solution-base" stored in ROMs and an imbedded search algorithm to retrieve the best solution. An appropriate search algorithm, called the virtual graphic method (VGM) has been developed. The VGM solves non-linear equations without performing arithmetic operations, and is therefore fast and easily implemented in hardware such as a VLSI. A flowchart and a conceptual diagram for VLSI implementation are presented.

INTRODUCTION

This paper proposes an approach for implementing mechanical CAD systems using electronic chips. It is based on the assumption that a potential mechanical product can be decomposed into basic elements. With this approach, the design of each element, prepared by experts, will be captured on a multi-purpose electronic chip. Examples of basic elements are primitive components such as pins and bolts, or more complex components such as beams and shafts. Potentially a vendor of mechanical CAD could create a customized system based on these available chips. Other electronic chips would be dedicated to the mating of parts to sets, and to matching sets into sub-assemblies (e.g. bolts and nuts, belts and sheaves to transmissions). This approach does not mean that software is eliminated in the proposed CAD system, but simply increases the reliance on hardware. There is thus a shift of emphasis, with a more equitable sharing of tasks between hardware and software.

The proposed idea can be implemented by software alone, and still has the advantage of capturing an expert's design. It can be utilized by designing basic mechanical elements either on a dedicated microprocessor or with the aid of a corresponding subroutine. The first method, however, leads to parallel processing and calls for the development of a special high-level language to coordinate the operations of the various processors. The second method increases the computational load on the main processor and slows the execution of the entire design procedure. In contrast, the approach proposed here reduces the execution time by one to three orders of magnitude[1] because hardware is faster than software. Moreover, by having many chips in parallel execution and by solving simultaneously for different mechanical elements, the total execution time for the solution of a design problem is again significantly decreased. The advantages of parallel computing are obtained while retaining a single processor and a regular programming language.

The hardware-based approach calls for the utilization of only limited sets of each available component type: for example, only certain bolts or certain discrete diameters of rods. The ability to bound a set of components is advantageous, not only because economic judgement calls for standardization, but mainly because industrial reality encourages limited inventories. Although standards limit the variations of the potential number and size of components, in practice availability is even further reduced since inventories cannot hold all the scope of a standard.

An analogy from the electronics industry is, for

*This research was supported by a NSF award for Novel Research, Grant #DMC-8604822.

example, where ICs of AND-gates come with 2, 4 and 8 inputs but not with 5, 6 and 7. Similarly, millimetric pins are sold in sizes 6, 8, 10 and 12 but usually not 7, 9 and 11; in our opinion, the number of available dimensions can be further reduced. Moreover, inventory restrictions are compelling, even for large companies. For example, the French aircraft company, Avions Marcel Dassault, decided some 25 years ago to restrict itself to only five different aluminum alloys.

The use of chips in mechanical CAD systems calls for a different approach to the design of basic mechanical components. This paper introduces such an approach and provides examples of VLSI chips for basic components. In conventional design, the required parameters x (e.g. a diameter of a shaft) are computed from given input variables V (e.g. a torque) according to a known relationship

$$x = f(V). \tag{1}$$

In the proposed approach, a set of standard x is precalculated as a function of V and stored in hardware (e.g. ROM). Later, when the CAD user specifies a certain V_c, the chip will retrieve a solution according to

$$V_0 = f^{-1}(x) \tag{2}$$

where f^{-1} is used symbolically to indicate that the chip conducts a search in the stored set x to find a solution V_0 which will be closest to V_c. Although the literature provides many known search methods,[2] another procedure, which can be easily implemented in hardware, will be proposed later in this paper. Expert knowledge must be incorporated into the design of the chip (which contains the search algorithm), in order to decide, for example, if $V_0 > V_c$ or if $V_0 < V_c$ is the acceptable solution.

In our approach, the chips involved do not perform any complex calculations (such as divisions, root extractions, etc.) but only control searches and make comparisons. Therefore, they are able to execute very rapidly.

SINGLE PARAMETER DESIGN PROBLEMS

Many design problems of primitive elements require the calculation of a single parameter (output) as a function of a given variable (input) and a certain material (i.e. given σ): for example, a knuckle joint, a chain link or a clevis joint. To facilitate the discussion, it is assumed that for all parts, σ is the maximum allowable design stress, τ is the shear stress such that $\tau = \sigma/2$ and the load is pure static tension (no bending).

Knuckle joint

The dimensions of the knuckle joint[3] shown in Fig. 1 are given as a function of d, which is the diameter of the tension rod (part 1) or of the shear pin (part 2). Theoretically, if loaded by a pure tensile force P, the tension rod and shear pin will fail simultaneously. It is the task of the designer to dimension the parts for functionality and for compliance with design constraints and specifications (e.g. loads, cost, weight, endurance).

Based solely on static strength considerations, d would be solved from the tensile stress equation

$$P = \frac{\pi}{4} d^2 \sigma \tag{3}$$

from which

$$d = \sqrt{\frac{4P}{\pi\sigma}} \tag{4}$$

For a force of 2500 N, for example, the following results would be obtained at different stress levels:

σ (N/mm^2)	127	255	637
d (mm)	5.00	3.53	2.24

Assuming only four diameters 1, 2, 4 and 6 are kept in stock, and that the design calls for a minimum diameter, then the sizes to be selected would be 6, 4 and 4. Obviously, the precise diameter often need not be calculated. What is important is the ability to retrieve a pin of appropriate diameter from the available stock.

To illustrate the development of the proposed approach, a simplified design issue will be discussed. Assume only four diameters (1, 5, 10 and 20 mm) and two materials (σ = 127 and 637 N/mm^2) are available. The limit load force P is calculated by Eq. (3) for each case, and rounded-off in Table 1.

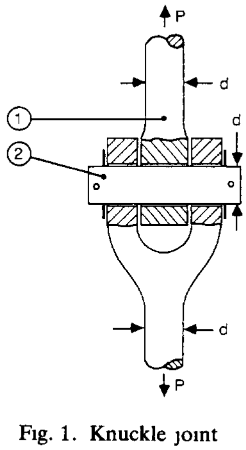

Fig. 1. Knuckle joint

Table 1. Limit loads P (in Newtons)

d (mm)	σ (N/mm^2) 127	637
1	100	500
5	2500	12,500
10	10,000	50,000
20	40,000	200,000

Table 3. Binary representation of d and σ

d (mm)	Binary representation a	b	σ (N/mm^2)	Binary representation c
1	0	0	127	0
5	0	1		
10	1	0	637	1
20	1	1		

With a system constructed after Table 1, the designer specifies P and σ, expecting the smallest feasible d in response. This design problem can be implemented in electronic hardware (i.e. VLSI) without the explicit use of Eq. (3) to avoid multiplication, division and square root extraction.

Two approaches for the design of specific VLSIs dedicated to the design of mechanical components will be introduced, neither of which is based on the emulation of software-based design. The first approach, based on a set of design rules arranged in what is called a "production system" in artificial intelligence,[2] is more elegant. These design rules are translated into a logic circuit, then implemented with logic gates into a VLSI to provide for the automatic selection of the appropriate solution. The second approach relies on tables of solutions (similar to Table 1) stored in ROMs, with automatic hardware scanners that search for the best solution and is more practical.

DESIGN RULES APPROACH

First, the design rules approach will be explained using the knuckle joint example. The design rules are derived from the eight load stages of Table 1. These loads represent the limits to the forces Pi, ranging from stage $i = 0$ to stage $i = 7$ (see Table 2).

The design rules approach requires that a pre-processor perform a sequential order representation of the variables involved in the design. For example, if only two materials are available, they are shown as 0 and 1. If only four diameters of pins are stocked, they are presented as 0, 1, 2 and 3, regardless of their real value. These numbers are processed and stored in the computer in their binary representation. For the knuckle joint example, the diameter representation of the four pins and the maximal stresses of the two given materials are shown in Table 3. Once the designer specifies a material, the corresponding binary representation is stored in the σ-register (see Fig. 2, to be explained further on).

Table 2. Load stages

Stage i	Pi range
0	$0 < P0 \leqslant 100$
1	$100 < P1 \leqslant 500$
2	$500 < P2 \leqslant 2500$
3	$2500 < P3 \leqslant 10{,}000$
4	$10{,}000 < P4 \leqslant 12{,}000$
5	$12{,}500 < P5 \leqslant 40{,}000$
6	$40{,}000 < P6 \leqslant 50{,}000$
7	$50{,}000 < P7 \leqslant 200{,}000$

When the designer specifies a load, the pre-processor compares it with each of the eight stages in Table 2 and stores the result (in the P-register) as a binary number corresponding to the appropriate stage. For example, if $P = 8000$ N is given, this is translated by the pre-processor to 3 and saved in the P-register as 011. The latter is decoded and passed to the design rules block in Fig. 2, as 00001000. Similarily, a load of 45,000 N, for example, is stored as 6 and transferred as 01000000.

A block diagram of the proposed hardware for the calculation of the pin diameter is shown in Fig. 2. The design rules block provides the relationship among the variables, and decides if a given combination of σ, d, and P is an "acceptable solution". The solution is acceptable if a pin of given material σ and diameter d will not shear under a specific load P.

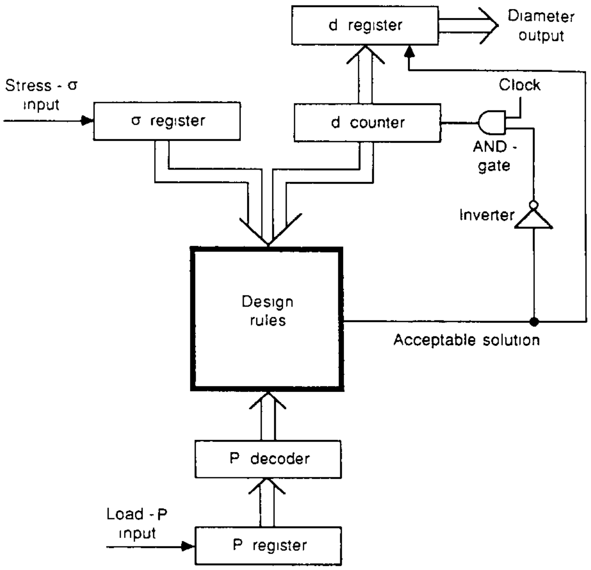

Fig. 2. Block diagram of a VLSI implementing design rules.

The operation of the hardware will now be explained. The material type is inserted in the σ-register, and the range of the required force is stored in the P-register. Inserting a new value for P will automatically reset the d-counter to 0 (not shown in Fig. 2). If the given force is less than 100 N, then the "acceptable solution" ("as") signal becomes 1. This signal is first inverted, and blocks the AND-gate at the input of the d-counter, and then loads the contents of the d-counter into the d-register, which, in turn, transmits this value as the diameter output.

Assume that a force $P = 8000$ N and stress $\sigma = 127$ N/mm^2 are given and entered in the respective input registers. The d-counter is reset automatically (to 0) causing the "as" signal to indicate 0. At the next clock pulse the counter is incremented to 1, but the "as" remains 0 (see Table 1). Only when the counter is incremented to 2 does the "as" signal become 1, thereby stopping the d-counter. The output of the VLSI is now 2, which is the equivalent of a pin with a diameter of 10 mm.

The cardinal part of this VLSI is the block containing the design rules, which must be determined for each case. The number of design rules for a knuckle joint is equal to the product of the quantity of obtainable types of materials multiplied by the amount of available pin diameters. In this example, the number of 2 × 4 = 8. The design rules can be derived with the aid of Karnough maps or any other technique. The design rules for this example can be stated as follows:

$P0 = 1$	(any d for any σ)
$P1 = (a + b) + c$	($d \geqslant 5$, or any d with $\sigma = 637$)
$P2 = a + b$	($d \geqslant 5$, independent of σ)
$P3 = a + bc$	($d \geqslant 10$, or $d = 5$ and $\sigma = 637$)
$P4 = ab + (a + b)c$	($d = 20$, or $d \geqslant 5$ and $\sigma = 637$)
$P5 = ab + ac$	($d = 20$, or $d \geqslant 10$ and $\sigma = 637$)
$P6 = ac$	($d \geqslant 10$ and $\sigma = 637$)
$P7 = abc$	($d = 20$ and $\sigma = 637$).

The variables a, b and c are defined in Table 3. The logic circuit that executes the eight design rules is shown in Fig. 3. The operation of this circuit is easy to follow, especially for the two extreme load stages. For load stage 0 ($0 < P1 \leqslant 100$, Table 2) the P-decoder shows a 1 only in cell #0 (right-most) so the output of AND-gate A now also becomes 1. As a consequence, the "as" output is 1 in compliance with the design rule $P0 = 1$.

For load stage 7 ($50{,}000 < P7 \leqslant 200{,}000$) the P-decoder has a 1 in cell #7 while all other cells are 0. Only if $d = 11$ (binary) and $c = 1$ the output of AND-gate K becomes 1; in turn the output of AND-gate H becomes 1. Subsequently, the "as" is 1. Evidently, the "as" becomes 1 for the largest diameter and the highest stress ($d = 20$, $\sigma = 637$), as required by the design rule $P7 = abc$.

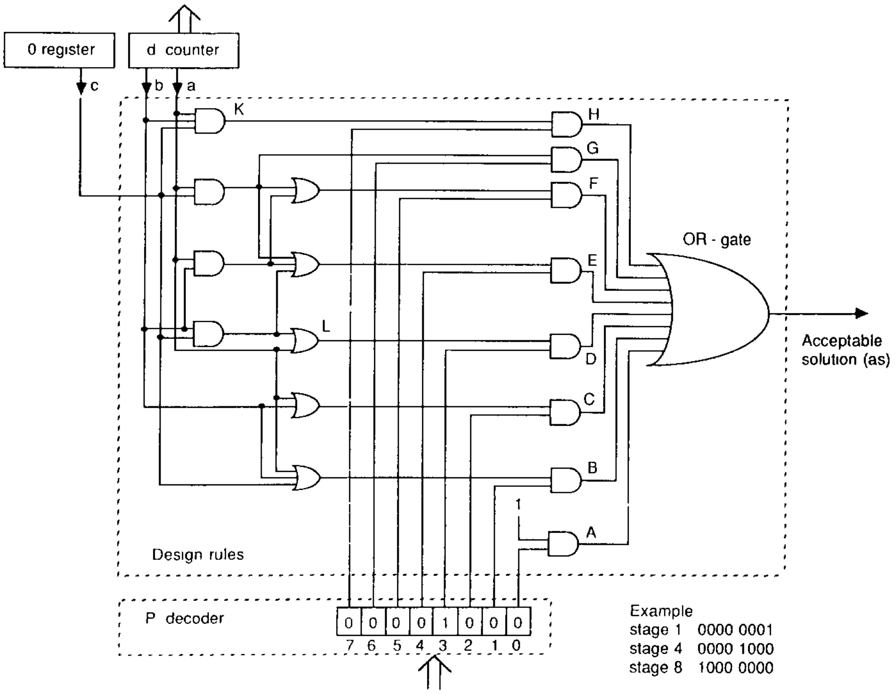

Fig. 3. Logic circuit detail.

As a further example, take $P = 8000$ N and $\sigma = 127$ (i.e. $c = 0$ from Table 3). Then, from Table 2, the load stage is $P3$ and cell #3 in the P-decoder is 1, as shown in Fig. 3. In the logic circuit of Fig. 3, when the d-counter reaches 10 (i.e. $a = 1$ and $b = 0$), the output of OR-gate L becomes 1, thus providing AND-gate D with a 1. At the same time, as stated before, the other input from the P-decoder is also 1 and, therefore, the output of AND-gate D is 1 and the "as" output becomes 1 as well.

Although the system shown works, its complexity for a great number of items results in many design rules; these grow substantially when more variables and more parameters are involved. In such cases, this method becomes questionable and another, more practical approach is needed. In the next section, problems with two independent parameters will be considered in a different and more practical way.

TWO-PARAMETER DESIGN PROBLEMS

The cantilever beam shown in Fig. 4 is an example of a basic element with a concentrated load P at the tip, stressing the beam in pure bending. The two input variables are the load P and the tip deflection δ. The beam has a circular cross-section of diameter d and length l, which are the two output parameters. The steel beam withstands a maximum tensile stress σ, and E is the elasticity modulus (Young's modulus).

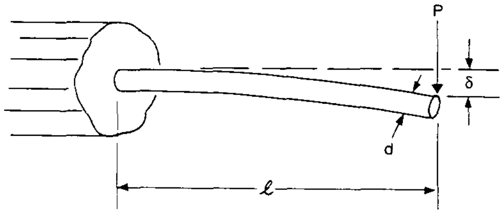

Fig. 4. Circular cross-section cantilever beam.

Cantilever beam analysis provides two equations[5] for beams of circular cross-section

$$\sigma = \frac{Md/2}{I} \tag{5}$$

and

$$\delta = \frac{Pl^3}{3EI}\cdot \tag{6}$$

Substituting the moment of inertia $I = \pi d^4/64$ and the bending moment $M = Pl$ into Eq. (5) yields

$$P = \frac{\pi\sigma d^3}{32l} \tag{7}$$

and substituting P from Eq. (7) into (6) yields

$$\delta = \frac{2\sigma l^2}{3Ed}\cdot \tag{8}$$

In many design problems, the length l and the diameter d of the beam have to be calculated for a given maximum load P and deflection δ. The problem now becomes one of solving a set of two non-linear equations with two unknowns. Solving Eqs. (7) and (8) for d and l yields

$$d = \sqrt[5]{\left(\frac{1536E\delta P^2}{\pi^2\sigma^3}\right)} \tag{9}$$

and

$$l = \sqrt[5]{\left(\frac{108PE^3\delta^3}{\pi\sigma^4}\right)}. \tag{10}$$

For example, assume the need to calculate d and l to support a load $P = 3$ N with a deflection $\delta \leqslant 2$ mm for $\sigma = 120$ N/mm^2 and $E = 21 \times 10^4$ N/mm^2. The solution of Eqs. (9) and (10) provides $l_0 = 129.8$ mm and $d_0 = 3.21$ mm. The designer can opt for the nearest available d, namely 3 or 4, and calculate the length l accordingly. In practice, however, he will probably round-off to the nearest higher diameter and keep the same length, thus obtaining $d = 4$ and $l = 130$.

In this particular cantilever beam design problem there is a closed-form analytical solution, but in general this might not be the case, and a graphical solution might be considered. Equations (7) and (8) can be rewritten to yield l as a function of d, and become, respectively:

$$l = \frac{\pi\sigma d^3}{32P} \tag{11}$$

and

$$l = \sqrt{\left(\frac{3E\delta d}{2\sigma}\right)}. \tag{12}$$

Equations (11) and (12) are plotted in Fig. 5 as lines A and B, respectively. Line A is the locus of all the sets of d and l (d, l), for which $P = 3$ N and the maximum stress is $\sigma = 120$ N/mm^2. Any set (d, l) above line A will satisfy $P > 3$ N. Similarly, line B is the locus of $\delta = 2$ mm and $\sigma = 120$ N/mm^2; all (d, l) above line B satisfy $\delta < 2$ mm.

The intersection of lines A and B indicates the length l able to support a load of $P = 3$ N with a deflection of $\delta = 2$ mm while utilizing the material most efficiently (i.e. to its maximum stress). This

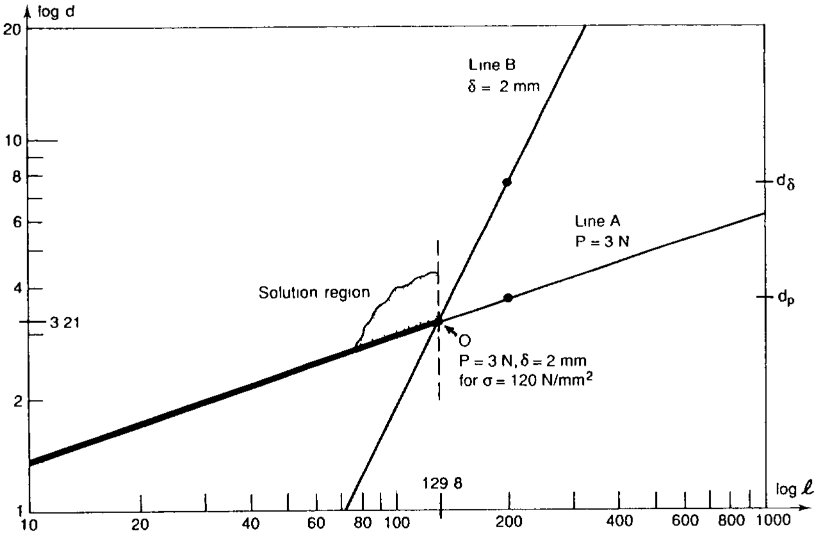

Fıg. 5 (*d*, *l*) vs (*P*, δ)

solution is identical to the arithmetic results obtained from Eqs. (9) and (10), namely $d_0 = 3.21$ mm and $l_0 = 129.8$ mm.

The graphical solution shown above was presented as an introduction to the operation of the hardware implementation, which will be explained later on.

According to the approach taken in this paper, pairs of solutions for the problem are precalculated and stored in a ROM. This means that for given discrete dimensions of d and l, pairs of results for the load P and the deflection δ are calculated for the maximal stress ($\sigma = 120\ \text{N/mm}^2$, for steel with $E = 210{,}000\ \text{N/mm}^2$). This is shown in Table 4 where pairs of P (in Newtons) and δ (in millimeters) with P at the top (e.g. 120.64 N) and δ underneath (e.g. 0.12 mm), are calculated by Eqs. (7) and (8) for sets of (d, l).

Table 4 represents two different but similarly organized ROMs. The first stores the loads P as functions of d and l, and the second retains the deflections δ for the same ds and ls as in the first ROM. This arrangement deals with one specific level of stress; similar tables are needed for various values of σ.

An appropriate search algorithm must be devised to select the proper set (d, l) satisfying $P \geqslant 3$ N and $\delta \leqslant 2$ mm for a maximum l and for efficient use of the material. A simple search through Table 4 for pairs of P and δ satisfying the conditions $P \geqslant 3$ N and $\delta \leqslant 2$ mm yields those pairs lying immediately above the "staircase" line drawn through the table.

Table 4 Cantılever beam loads and deflections

				l (mm)				
d (mm)	50	75	100	125	150	200	250	300
8	120 64 0.12	80.42 0.27	60.31 0.48	48.25 0.74	40.21 1 07	30.16 1 90	24.13 2.98	20.11 4.29
6	50.89 0 16	33.93 0.36	25.45 0.63	20.36 0.99	16.96 1.43	12.72 2.54	10.18 3.97	8.48 5.71
4	15 08 0 24	7.54 0.95	7.54 0.95	6.03 1.49	5 03 2.14	3.77 3.81	3 02 5.95	2 25 8.57
3	6.36 0.32	4 24 0.71	3 18 1.27	2 54 1.98	2.12 2.86	1 59 5.08	1.27 7.94	1 06 11 43
2	1 88 0.48	1.26 1 07	0 94 1 91	0 75 2.98	0.63 4.29	0.47 7.62	0.38 11 90	0.31 17.14
1	0.24 0.95	0 16 2 14	0.12 3.81	0.90 5 95	0.08 8.57	0.06 15 24	0.05 23.81	0.04 34 29

Now the problem becomes one of deciding which of those pairs satisfies the other two conditions, namely maximum length and best use of the material.

THE VIRTUAL GRAPHIC METHOD (VGM)

It is already known (from Fig. 5) that the solution is located above line A and to the left of the intersection point O. However, Table 4, which maps the plan (d, l) (see Fig. 5), does not display any intersection. To overcome this difficulty we propose to introduce a virtual graphical method (VGM), which will not only indicate the neighborhood of point O, but will also allow the retrieval of the better set (d, l).

Any arbitrary vertical line intersects the lines A and B at points d_P and d_δ, respectively. For a vertical on the right-hand side of point O, $d_p < d_\delta$. For example, for $l = 200$, $d_P = 3.7$ and $d_\delta = 7.6$ (see Fig. 5). Similarly, for any vertical satisfying $d_P > d_\delta$, the vertical passes to the left-hand side of O. Hence, the inequality displays a change of sign at the cross over from one side of the intersection to the other.

The virtual graphic method (VGM) will now be demonstrated with an example of the search for a solution in d and l. Starting from the lower right-hand corner in Table 4, the search for both $P \geqslant 3$ N and $\delta \leqslant 2$ mm will proceed column-wise (from the smallest toward the larger d and then from the longest l through the shorter ones). The search proceeds towards the left, favoring solutions on line A, for which the slope is one-sixth of that for line B. The diameter d benefits from reduced sensitivity to small changes in length (for line B, small variations of length generate large differences in diameter). From the bottom of the right-most column ($l = 300$) up, the lowest acceptable value for P (8.48) is reached for diameter $d_P = 6$. In the same column, no value is found for $\delta \leqslant 2$; therefore, the search regresses to the previous column ($l = 250$). Now the smallest suitable P (3.02) appears at $d_P = 4$, but again there is no valid δ; hence, the column $l = 200$ is scanned next.

This time the condition $P \geqslant 3$ is met at $d_P = 4$ ($P = 3.77$), but the condition $\delta \leqslant 2$ is fulfilled at $d_\delta = 8$ ($d = 1.90$), namely, $d_P < d_\delta$ ($4 < 8$). Stepping back one column ($l = 150$), $d_P = 4$ ($P = 5.03$), and the deflection (1.43) yields $d_\delta = 6$. Again, $d_P < d_\delta$ ($4 < 6$), and the search continues.

One more column backwards ($l = 125$) shows $d_P = 4$ ($P = 6.03$) and $d_\delta = 3$ ($\delta = 1.98$). Now $d_P > d_\delta$, indicating that the intersection of lines A and B has just been crossed and the search ends. The first solution to the left of the intersection and above line A (see Fig. 5) has been reached for $d = 4$ and $l = 125$.

A flowchart for the VGM search algorithm is shown in Fig. 6. The lengths are denoted as l_j, with $j = 0,1,2, \ldots, j_{\max}$, and the diameters are designated d_i, with $i = 0,1,2, \ldots, i_{\max}$. By the same notation convention, the load for diameter i and length j becomes P_{ij}, and the counterpart deflection is written as δ_{ij}.

After entering the given data (P, δ), the search starts at maximum length l ($j = j_{\max}$) and smallest diameter d ($i = 0$). As long as the first condition (Is $P_{ij} > P$?) is not satisfied the upper i-loop is incremented. Once this condition is satisfied, the corresponding diameter is stored as d_p, the i-counter is reset, and the search for d_δ starts. The search for d_δ is identical to that for d_p. If the second condition (Is $\delta_{ij} < \delta$?) fails, then the middle i-loop is incremented. As before, if the condition is met, the diameter is retained as d_δ and the i-counter is reset. Now both diameters are compared (Is $d_P < d_\delta$?), and if the condition is satisfied, the solution is obtained ($d = d_P$, $l = l_j$). Otherwise the length-counter is decremented ($j = j - 1$) and the search for d_P and d_δ continues.

If an upper limit d is reached (Is $i = i_{\max}$?) without fulfilling either one or both of the P and δ conditions, then the search escapes to the next shorter length l ($j = j - 1$). No solution exists if all lengths down to the shortest (Is $j = 0$?) prove unsatisfactory.

A block diagram of the main functions of the corresponding digital chip is shown in Fig. 7. The values of P and δ are given, and the ROMs for a certain stress σ are searched in the manner shown previously for Table 4. The left-hand side of Fig. 7 is designed to search for d_P and the right-hand side is structured to retrieve d_δ. The middle area searches for the length l and retrieves the solution. The given P is entered at left and compared to the first load value (extracted from ROM P). If "$P_{ij} > P$?" is not satisfied, then the "d_P up-counter" is incremented through the successive diameters available until the condition is met. Simultaneously, the same process occurs with the deflection condition (Is $\delta_{ij} < \delta$?) for d_δ. After d_δ is found, it is compared to d_P (Is $d_P < d_\delta$?). At this point either a solution is found or the search will continue by resetting the d-counter and decrementing the l-counter. When solutions are found, they are passed to the d_P and l registers. To simplify the diagram, the circuit in Fig. 7 contains only the main functions; other functions, such as recovery from saturation of the d-counters and the l-counter, are not shown.

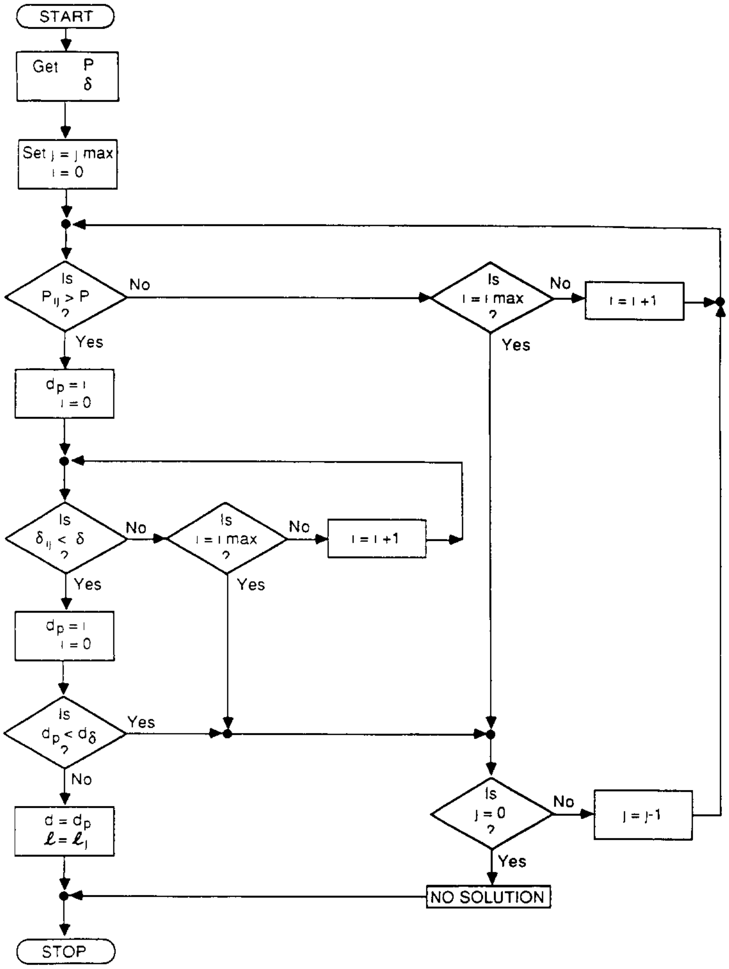

Fig. 6. Flowchart of the search for $P \geqslant 3$ N and $\delta \leqslant 2$ mm

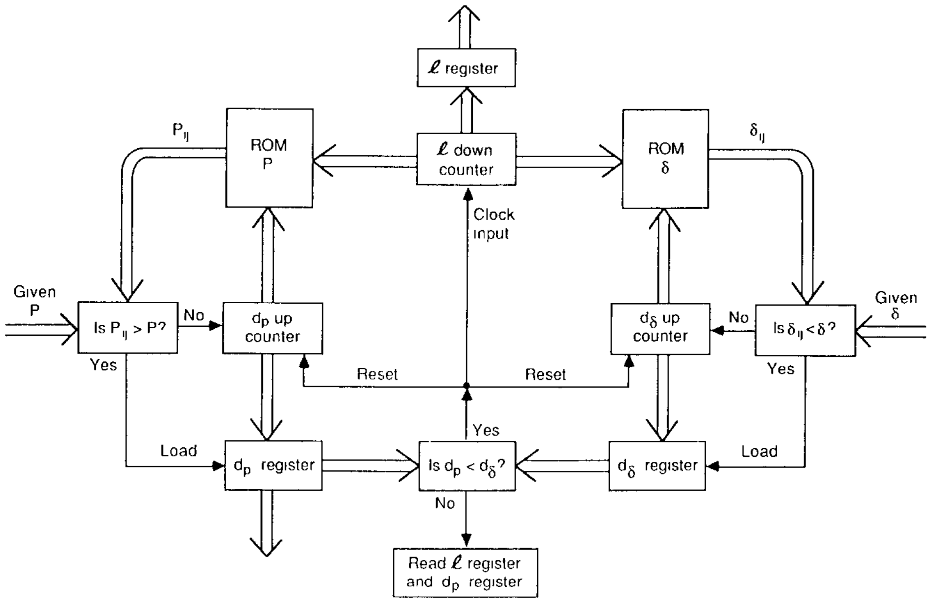

Fig 7 Hardware architecture of the search element (main functions)

VLSI IMPLEMENTATION

The circuit in Fig. 7 actually solves two non-linear equations with two unknowns without looping through any iterations. Furthermore, no higher order arithmetic operations have been performed (such as root extraction, exponents or even division or multiplication). The result is achieved by searching and comparing values. As there is no need to perform arithmetic operations or consecutively store results, the response occurs very quickly.[6]

The circuit in Fig. 7 is actually composed of two parts, shown in Fig. 8: a search element and a ROM which stores the potential solutions to the problem. The search element can solve a wide class of two non-linear equations with two unknowns, which do not necessarily have a closed solution as in the example presented above. This justifies production of the search element as a VLSI or LSI chip.

The strategy used for controlling the search might be critical in determining the correct solution. The nature of the problem at hand dictates the type of search to be applied. Therefore the search element is provided with pins controlling the direction of the search. It is the responsibility of the expert who designs the CAD system to select the appropriate search algorithm by connecting these pins to "1" or "0" level logic.

The search can also be restrained to look for the solution in a bounded field of parameter constraints (for example $l_{\min} < l < l_{\max}$). This can be executed by setting lower and upper bounds on the associated counters through additional pins in the search element. Moreover, these bounds can be set automatically using the results obtained from other VLSI chips involved in the computation, provided the components under design are related (e.g. a pair of gears).

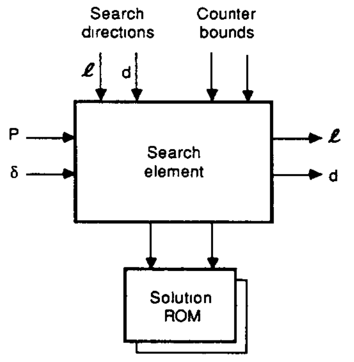

Fig. 8. Conceptual diagram of the search element.

By having many VLSI chips in parallel execution and by solving simultaneously for many components of a mechanical product, the total execution time for solving the product design problem is significantly reduced. The proposed VLSIs are even more advantageous than corresponding software subroutines when the equations solved do not have a closed solution. Solving by software requires many iterations and is time-consuming, whereas the search method always requires the same computational effort.

Additional significant advantages of using VLSIs are the low acquisition price of a new CAD system and the flexibility to enhance and upgrade. VLSI chips are inexpensive when produced in quantity and compared to the price and maintenance costs of software packages. Selection of the appropriate VLSIs will allow the assembly of customized CAD systems without the loss of flexibility, ease of retrofit or expansion. Another advantage of VLSIs is that they are tamper-proof, thereby excluding the possibility of unintentional damage.

CONCLUSIONS

This paper demonstrated the feasibility of applying generic VLSIs as an aid to the design of mechanical components. The VLSI consists of a search circuit which finds the near-optimal result in a solution-base stored in a ROM. The final output is not optimal, but the solution is highly satisfactory and falls within the neighborhood of the optimum. This method is appropriate for practical purposes and for technical design decisions. It is very quick, efficient and hence also helpful for the conceptual and preliminary design phases.[8] It incorporates expert design knowledge, the reduction of computer execution time and a self-imposed restriction to a limited set of available components.

REFERENCES

1. Leung, S., Shanblatt, M.A.: A VLSI chip architecture for the real-time computation of direct kinematics. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1717–1722, San Francisco, CA, 1986.
2. Rich, E.: *Artificial Intelligence*. New York, McGraw-Hill, 1983.
3. Baumeister, T. (Ed.): *Mark's Standard Handbook for Mechanical Engineering*, 8th Edn, pp. 8–35. McGraw-Hill, New York, 1978.
4. Kohavi, Z.: *Switching and Finite Automata Theory*. New York, McGraw-Hill, 1978.
5. Timoshenko, S.: *Strength of Materials, Elementary Theory and Problems*, Vol. I. Scarborough, Van Nostrand, 1955.

6. Mead, C., Conway, L.: *Introduction to VLSI Systems* Menlo Park, Addison-Wesley, 1980.
7. Dixon, J.R., Simmons, M.K., Cohen, P.R.: An architecture for application of artificial intelligence to design. *Proceedings of 21st Design Automation Conference*, pp. 634–640, 1984.
8. Jakiela, M., Papalambros, P., Ulsoy, A.G.: Programming optimal suggestions in the design concept phase: application to the beothroyd assembly charts. *ASME J. Mechs. Transmissions Aut. Design* **107**: 285–291, 1985.