C O M M O N W E A L T H of A U S T R A L I A

PATENTS ACT 1952

PROVISIONAL SPECIFICATION

for an invention entitled:

AN APPARATUS FOR USE IN PRODUCING A TIMETABLE

This invention is described in the following statement:

AN APPARATUS FOR USE IN PRODUCING A TIMETABLE

The present invention relates to an apparatus for use in producing a timetable.

The production of a timetable for such matters as classes, lectures or airline flights is an arduous task which needs to be completed without allocating people or resources more than once to a given time period within the timetable. There are also invariably a number of other constraints which need to be met, such as ensuring a particular class or flight occurs a minimum number of times within the timetable and ensuring teacher or pilot allocation limits are not exceeded. A timetable is normally set to cover a given length of time, such as a week or month, and is repeated when the length of time expires.

Timetables are normally established manually by a person who, having been assigned the task, first establishes the requirements which the timetable needs to meet. For example, a weekly school timetable may require a year 7 class to be taught English seven times a week, physical education two times a week, etc. The situation becomes complex, however, when students are members of different classes depending on subjects being taught, there are only a limited number of rooms available, established teacher workload criterias need to be met, and certain subjects require contiguous multiple periods to be allocated thereto. A satisfactory solution also requires the timetable to include no clashes, such as a period of the timetable having a teacher, class, or room allocated more than once to that period. After an initial timetable has been drawn up which satisfies all of the requirements and constraints, the timetable needs to be rearranged to remove all of the clashes. In some cases, depending on the requirements and constraints involved, a no clash solution appears impossible

to achieve.

Software packages are available to assist a scheduler who has been assigned the task of establishing a timetable, however, the packages require substantial input from the scheduler. One package only provides an indication of where clashes occur in a timetable produced by the scheduler and another package is only able to construct a timetable after is it provided with blocks of classes which must be taught concurrently.

An object of the present invention is to provide an apparatus which is able to produce satisfactory timetable solutions with a minimal amount of human input.

In accordance with the present invention there is provided an apparatus for use in producing a timetable which allocates elements, such as people, resources and periods, to units of the timetable, said periods being within a predetermined length of time of said timetable, said apparatus comprising:

first means for storing timetable signals representative of an initial timetable with said units having preselected elements;

means for accessing one of said units in said first storing means and selecting an element of the accessed unit and a change for the selected element;

second means for storing cost signals for respective elements, said cost signals being representative of whether the elements satisfy predetermined element criteria, said second means being responsive to said accessing means and said first means to access said cost signals for respective elements in parallel, and generate respective change in cost signals representative of the effect of said change on said timetable; and

means responsive to said change in cost signals for determining whether to accept said change and if said change

is accepted, for accessing said first means to alter said timetable signals to effect said change.

Preferably said second means stores said cost signals in duplicate for said respective elements and said second means accesses said duplicate cost signals, in parallel, to generate first change in cost signals representative of deleting said selected element and second change in cost signals representative of inserting said change.

Preferably an element signal representative of the value of said selected element and a change signal representative of the value of said change, are used to access said duplicate cost signals, respectively, for the remaining elements of said accessed unit to generate said change in cost signals.

Preferably said selected element signal and a further element signal representative of the value of a further element of said accessed unit are used to access one set of said duplicate cost signals for the further element to generate said first change in cost signals and the other set of said duplicate cost signals for the further element are accessed by said change signal and said further element signal to generate said second change in cost signals.

Preferably said determining means derives a total cost change value from said change in cost signals and if said total cost change value is less than a predetermined level causes said change to be accepted. Preferably said determining means further includes means for applying an annealing algorithm to said total cost change signals if said total cost change signals are greater than said predetermined level, to determine whether to accept said change.

Preferably said total cost change value may be derived

from a plurality of change in cost signals generated after accessing a plurality of said units and selecting a plurality of changes for said selected element.

Preferably said predetermined criteria includes people and resources are not to be allocated more than once to one of said periods.

A preferred embodiment of the present invention will hereinafter be described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram of an apparatus for use in establishing a timetable;

Figure 2 is a flow diagram illustrating the state transitions of a finite state machine of the apparatus; and

Figures 3 to 9 are detailed circuit diagrams of the circuitry of the apparatus.

An apparatus 2 for use in producing a timetable, as shown in Figure 1 includes a timetable memory circuit 4, a clash array circuit 6, a cost computation circuit 8, an access circuit 10, a control circuit 12 and a pc (personal computer) interface 14. The circuits of the apparatus 4 are mounted on a pc circuit board which is connectable to a standard IBM pc bus via the interface 14. Once connected to the bus, the host computer is able to read and write data to and from memory elements of the memory circuit 4 and the array circuit 6. The control circuit 12 is implemented by a dedicated finite state machine and controls the transfer of data between the remaining circuits of the apparatus 2. Control lines and connections between the control circuit 12 and the pc interface 14 and the remaining circuits 4, 6, 8 and 10 are omitted for clarity.

The timetable memory circuit 4 is adapted to store an array of records 16 which represent a required timetable. The timetable may be for an educational institution, such as

a school or university or for an organisation, such as an airline, which needs to allocate its personnel and resources to a repetitive schedule. The timetable may be for any given length of time, such as a week or month or yearly quarter. A record 16 of the memory circuit 4 includes a plurality of fields 18 which each represent an element of a unit of the timetable. For a class or lecture, the elements includes a teacher, a class number and a period, whereas for an airline, the elements may include flight crew, service crew, and flight number. The apparatus 2 illustrated in Figure 1 has been implemented specifically for producing a timetable for an educational institution, such as a school, and the remainder of the description is directed to production of a school timetable. It will be apparent, however, that the apparatus 2 can be simply modified to enable generation of timetables for other organisations.

An initial timetable is stored in the timetable memory circuit 4 and the units are stored as records 16 and each include the following fields 18:

1. A teacher field 20 which includes a number representative of a particular teacher.

2. A class field 22 which includes a number representative of a particular class. Although most of the students belonging to a class do not belong to another class, a student may belong to more than one class and those classes are considered to clash, if allocated to the same period, as discussed hereinafter.

3. A period field 24 which includes a period number. The timetable stored in the circuit 4 is a weekly timetable and the periods are hourly, 50 minute or 45 minute, etc. time slots within the week during which a particular class is conducted. The periods are allocated a number between 0 to 127. If a day is only

to include 6 periods, 8 or 16 period numbers are allocated to Monday but only period numbers 1 to 6 are considered relevant and the remaining period numbers 7 to 16 say, are ignored. Similarly for Tuesday, only period numbers 17 to 22 are considered relevant and the remaining period numbers 23 to 32 are ignored. This provides flexibility with respect to the allocation of periods and removes the requirement for a division unit as modulus operations can be performed on the period numbers to determine which day they relate to instead of performing a division operation on the numbers.

4. A room field 26 which includes a number representative of a particular room in which a class is to be conducted.

5. A subject field 28 which represents a particular subject to be taught to the class, such as Maths or English.

6. A class group field 30 which indicates to which class group the class represented by field 22 belongs to.

7. A teacher group field 32 which indicates the teacher group which the teacher represented by field 20 belongs to.

8. A preference field 34 which includes a memory pointer to a list including a preferred period number or numbers to which a respective unit may be allocated.

9. A multiple period pointer field 36 which includes an address for accessing a partner location memory 40 of the memory circuit 4. The partner location memory 40 provides the period number of the unit with which the access unit is to form a multiple period. Multiple

periods relate to two units which are allocated or need to be allocated to consecutive periods.

10. A multiple period flag field 38 which is set when the respective unit is a part of a multiple period.

11. A general flag field 40 which includes flags indicating a number of specific features of the respective unit, such as whether the end of the timetable has been reached, whether the current unit is part of a collection of linked units, whether the unit relates to the end of a teacher chain.

12. A chain field 42 which includes a pointer to the address of the next record 16 in the circuit 4 for which a teacher change is to be evaluated. The field is not employed if only period changes are evaluated.

For the initial timetable stored in the timetable memory circuit 4, all of the units include classes, teachers, rooms and subjects allocated as desired, including multiple periods, however, whether units clash or not with one another is disregarded. A basic clash occurs when an element of a unit is allocated more than once to the same period. For example, a class clash occurs if two units include class 1 in their respective class fields 22 and the units are both allocated to period number 1 in their respective period fields 24, notwithstanding that the room and teacher allocations may be different. An objective cost function can be evaluated for a timetable to represent the number of clashes which occur in a timetable. An acceptable timetable is one having a cost of zero, i.e. no clashes, and the function of the apparatus 2 is to minimise the cost of the timetable stored in the circuit 4.

The access circuit 10 proposes changes to the units of the timetable stored in the circuit 4 and the clash array

circuit 6 evaluates, in parallel, the effect of a proposed change on all of the elements of an accessed unit and generates change in cost signals which represent the cost change as a result of the proposed element change. The change in cost (ΔC) data is processed by the cost computation circuit 8 to determine whether to accept or reject the element change. If the element change is accepted, the corresponding record 16 in the memory circuit 4 is altered.

The access circuit 10 of the apparatus 2 is configured to propose random changes to either the period field 24 or the teacher field 20 of records 16 accessed by and outputted from the memory circuit 4. The memory circuit 4, under control of the control circuit 12, accesses records 16 sequentially during a period swap routine or according to a teacher chain during a teacher swap routine. The access circuit 10 includes a random number generator 50, a modulus memory circuit 52, a class clash list circuit 54 and two crossbar switches 56 and 58 which are all controlled by the control circuit 12. The control circuit 12, as shown in Figure 2, is initially in a idle state 100 and enters a single cycle wait state 102 after receiving signals instructing the apparatus 2 to begin execution of a swap routine. At state 102, the control circuit 12 examines status registers to determine whether the user wishes to run a teacher or a period swap routine. If period changes are to be performed, the circuit 12 proceeds to state 104 otherwise it proceeds to state 106, as shown in Figure 2. At both states 104 and 106, the control circuit 12 accesses the first records 16 in the memory circuit 4 and the contents of all of the fields, except the chain and flag fields 38, 41 and 42 and the teacher group field 32 are outputted in parallel to the clash array circuit 6.

Simultaneously, the random number generator 50 outputs a random number signal on line 60 which causes the modulus

memory circuit 52 to output a random teacher or period number on line 62. The random teacher or period number is used as a proposed teacher or period change for the accessed element. The modulus memory 52 is divided into two halves, one including valid period numbers and the other including valid teacher numbers. The control circuit 12 ensures the correct half is accessed depending on whether a teacher or period swap is to be performed. The modulus memory 52 ensures the number outputted on line 62 represents a valid teacher or period change in response to the number outputted by the random number generator 50. The modulus memory 52 receives the contents of the teacher group field 32 for the accessed unit so that only teacher numbers which form part of the group indicated by the field 32 are outputted on line 62. Selection of a teacher change occurs during state 108 of the circuit 12, which follows state 106.

The clash array circuit 6 includes a plurality of array circuits 70, which are memory circuits arranged so as to be accessed in parallel by, in most cases, a respective field 18 of an accessed record 16 and a teacher or period change outputted on line 62. The address lines of the array memory circuits 70 are connected to respective output lines 72 of the timetable memory circuit 4 and to at least one of the outputs of the crossbar switches 56 and 58. The contents of the fields 22, 26, 28, 30, 34 and 36 of the currently accessed unit are placed on the outputs line 72. As well as the selected change outputted on line 62, the contents of the teacher field 20 is inputted to the first crossbar switch 56 and the contents of the period field 24 of the accessed unit is inputted to the second crossbar switch 58. The teacher crossbar switch 56 connected to the teacher field output line 74 has an upper and lower output line 76 and 78, respectively, and the period crossbar switch 58 also has upper and lower output lines 80 and 82, respectively. The upper and lower lines of the crossbar switches 56 and 58 respectively receive the accessed field 20, 24 and the

selected change placed on line 62, or visa versa, depending on the state of the switches 56 and 58.

It will be apparent that the lines referred to in the present specification may comprise a serial data line or a parallel data bus, as desired.

The array circuits 70 store data which enables cost and requirement evaluations to be made with respect to the effect the selected or proposed change will have on respective elements of the accessed unit. An array circuit 70 may enable an evaluation to be made with respect to a period change or a teacher change or both. For most of the elements of the units stored in the timetable memory circuit 4, a first memory circuit is provided to enable the evaluation of the effect of changing or moving the element from its present status and another memory circuit is provided for enabling the evaluation of the effect of changing or moving the element to the proposed new status, as selected by the output 62 of the modulus memory 52. The data lines of the array memory circuits 70 are, in most cases, connected to respective arithmetic units, such as an increment/decrement unit 84 and/or a subtractor 86. The increment/decrement units 84 are used to effect the proposed changes or moves for the respective elements of a memory circuit 70 and to perform other changes in accordance with a swap routine of the control circuit 12. The increment/decrement units 84 also pass change in cost data determined from the respective memory circuits 70 onto a cost bus 88 of the array circuit 6. The subtractors 86, as is explained hereinafter, have outputs which pass change in cost data to a change in cost (Δc) bus 90. The buses 88 and 90 are connected to the cost computation circuit 8. Initial data for the array circuits 70 is stored when the initial timetable is entered.

For analysing the school timetable stored in the memory

circuit 4 with records 16 having the fields 18 described previously, the clash array circuit 6 includes the following array circuits 70.

A Class From memory circuit 92 includes records of how many times a particular class is allocated to a period. The memory locations of the circuit 92 are accessible by a class number from the field 22 of the accessed unit and a period number from the lower output line 82 of the crossbar switch 58. Each location stores the number of times the accessing class number has been allocated to the accessed period number. A Class To memory circuit 94 is the same as the Class From circuit 92 and has the same content as the From circuit 92 when the control circuit 12 is in state 104, as shown in Figure 2, at the beginning of a period swap routine. The Class To circuit 94, however, has address lines connected to the upper line 80 of the period crossbar switch 58 instead of the lower line 82. At state 104, the period crossbar switch 58 is set so the contents of the period field 24 is outputted to the lower line 82 and the selected period change is outputted to the upper line 80. The contents of the class field 22 is outputted on line 91 to the class circuits 92 and 94 which are read when the control circuit 12 enters state 110 following state 104. The number of times the accessing class has been allocated to the accessing current period stored in field 24 is outputted from the Class From circuit 92 and is decremented by the increment/decrement unit 84 connected thereto so as to provide a value which represents the decreased allocation which would occur if the accessing class is moved from that period. The new allocation is evaluated at the increment/decrement unit 84 to determine whether moving the class from the accessing period would involve a cost saving or not. A cost saving is only incurred if the class allocation to the accessing period is decreased from 2 to 1. The output line from the increment/decrement unit 84 to the cost bus 88 provides a signal which indicates whether a cost

saving or not would occur as a result of moving the accessing class from the accessing period. During the succeeding state 112 of the control circuit 12 the result of the decrement operation is written to the accessed location in the Class From circuit 92.

The simultaneous operations performed using the Class To circuit 94 are similar, except as the accessing period is the proposed change outputted by the modulus memory 52, the contents of the accessed location is incremented to reflect the increased allocation if the accessing class is allocated to the accessing period, by virtue of the proposed change. The value of the increased allocation is then examined by the increment/decrement unit 84 to determine whether it represents a cost increase or not, and a signal representative of the evaluation is outputted to the cost bus 88. A cost increase occurs whenever the value of the accessed location is increased beyond 1, i.e. a cost increase does not occur if the value only increases from 0 to 1.

The Class From and Class To circuits 92 and 94, together with the corresponding arithmetic units 84, enable the generation of signals, in parallel, which represent the change in cost which will occur if the class of an accessed unit is moved from its presently allocated period, and the change in cost involved in moving the class to the proposed period change. The circuits 92 and 94 and the units 84 store and provide class clash information or cost data which can be used in deciding whether to accept the proposed period change.

Similar circuits with respective increment/decrement units 84 are provided for the evaluation of the effect of the proposed change on teacher clashes and room clashes, being a Teacher From memory circuit 200, a Teacher To memory circuit 202, a Room From memory circuit 204 and a Room To

memory circuit 206. The From circuits 200 and 204 have address lines connected to the lower output line 82 of the period crossbar switch 58 and the To circuits 202 and 206 have address lines connected to the upper output line 80 of the period crossbar switch 58. The Room circuits 204 and 206 also have address lines connected to a room line 208 on which the contents of the room field 26 of the accessed unit is placed. The Room circuits 204 and 206 include memory locations which record the number of times a room is allocated to a particular period. The Teacher From circuit 200 also has address lines connected to the lower output line 78 of the teacher crossbar switch 56 and the Teacher To circuit 202 has address lines connected to the upper output line 76 of the teacher crossbar switch 56. When a period swap routine is being performed the teacher crossbar switch 56 outputs the contents of the teacher field 20 of the accessed unit to both output lines 76 and 78. When a teacher swap routine is being executed, and the control circuit 12 is in a state 114 corresponding to state 110 of the period swap routine, the period crossbar switch 58 is set so the upper lines 80 and 82 both receive the contents of the period field 24, and the teacher crossbar switch 56 is set to output the teacher field 20 and the proposed teacher change on the upper and lower output lines 76 and 78, respectively. The Teacher circuits 200 and 202 both include memory locations which record the number of times a teacher has been allocated to a particular period. The outputs from the From and To circuits 200 to 206 to the cost bus 88 also represent cost savings and increases as for the Class From circuit 92 and Class To circuit.

A Day Class Subject From memory circuit 210, a Day Class Subject To memory circuit 212, a Day Teacher From memory circuit 214 and a Day Teacher To memory circuit 216 are provided to evaluate the effect a proposed period change has on acceptable limits for the number of times a class may take a subject in a day and the number of times a teacher

may teach in a day, respectively. The From circuits 210 and 214 have address lines connected to the lower output line 82 of the period crossbar switch 58 and the To circuits 212 and 216 have address lines connected to the upper output line 80, as is the case for the Class From and Class To circuits 92 and 94. The circuits 210 to 216 include respective increment/decrement units 84 with the accessed contents of the From circuits 210 and 214 being decremented during state 110 and the accessed contents of the To circuits 212 and 216 being incremented during state 110. The Day Class Subject circuits 210 and 212 also have address lines connected to the class field output line 91 and the output line 218 which receives the contents of the subject field 28 of the accessed unit. The memory locations of the Day Class Subject circuits 210 and 212 store records of the number of times the accessing class is taught the accessing subject on the day of the accessing period number. Both of the outputs from the increment/decrement units 84 of the circuits 210 and 212 are two bit lines which enable signals to be outputted to the cost bus 88 which are representative of three or four conditions, instead of two conditions as discussed previously. Therefore, the proposed changes to the accessed allocations of the circuits 210 and 212 may be represented as satisfactory, not satisfactory, or highly satisfactory to the cost bus 88. The Day Teacher circuits 214 and 216 also have address lines connected to the lower output line 78 and the upper output line 76, respectively, of the teacher crossbar switch 56, and record the number of times an accessing teacher has been allocated to teach on the day of the accessing period number. The output lines to the cost bus 82 from the respective increment/decrement unit 84 of the circuits 214 and 216 may also be two bit lines.

Further circuits which relate to monitoring the effect on teacher allocations are a Class Group Teacher From circuit 220, a Class Group Teacher To circuit 222, a Teacher

From circuit 224, and a Teacher To circuit 226, which each include respective increment/decrement units 84. The circuits 220 to 226 are not used during period swap routines but are only used during teacher swap routines when they are accessed during state 114. The From circuits 220 and 224 have address lines connected to the lower output lines 78 of the teacher crossbar switch 56 and the To circuits 222 and 226 have address lines connected to the upper outlet line of the teacher crossbar switch 56. The Class Group circuits 220 and 222 also have address lines connected to the output line 228 which receives the contents of the Class Group field 30 of the accessed unit. The Class Group circuits 220 and 222 record the number of times an accessing teacher teaches an accessing class group number and again the output from the respective increment/decrement units 84 may be a two bit line to reflect how the proposed change relates to acceptable limits for allocation of teachers to class groups. The Teacher Allocation circuits 224 and 226 merely record the number of times an accessing teacher has been allocated to classes over the period of time covered by the timetable. The circuits 224 and 226 therefore present data to the cost bus 88 via one or two bit lines, which represent the effect on teacher allocation limits in deleting one teacher and inserting another teacher, respectively.

Array circuits 70 which do not include increment/decrement units 84 are a Preference From To memory circuit 230, a Multiple Period From circuit 232 and a Multiple Period To circuit 234. The Preference From To circuit has address lines connected to both the upper and lower output lines 80 and 82 of the period crossbar switch 58, and to a preference output line 236 which receives the contents of the preference field 34 of the accessed unit from the timetable memory circuit 4. The Preference From To circuit 230 stores a preference list of preference values and is accessible by the pointer stored in the preference field 34. The preference values each represent the cost of

having the accessed unit allocated to a respective period, which depends on whether the period is preferred or not. A unit may have more than one preferred period. The circuit 230 outputs the preference value for the accessing current period, and the preference value for the accessing proposed new period. The two values are outputted to a subtractor circuit 86, which outputs a signal representative of the difference between the values to the change in cost bus 90 which indicates whether the current period or the proposed new period is closer to or at a preferred period location. The signal outputted to the change in cost bus is either high or low depending on whether the difference evaluated by the subtractor 86 is negative, or greater than or equal to zero.

The multiple period circuits 232 and 234 are addressed by the period numbers placed on the upper and lower output lines 80 and 82, respectively, of the period crossbar switch 58. Circuits 232 and 234 are also addressed by the period number of the multiple period partner which is outputted on output line 238 of the partner location memory 40 described previously. Both circuits 232 and 234 produce an output signal which is placed on the cost bus 88 and the signal of circuit 232 provides an indication as to whether changing the current period breaks a multiple period grouping or not and the signal of circuit 234 represents whether making the proposed period change on the output line 80 creates a multiple period grouping or not.

During state 112 the change in cost signals placed on the cost bus 88 are inputted to a cost table circuit 240 of the cost computation circuit 8. The cost table circuit 240 applies appropriate weighting to the inputted signals to generate a weighted change in cost signal which is placed on the change in cost bus 90, which in turn, is connected to the input of an adder circuit 242 of the cost computation circuit 8. The cost table circuit 240 is preferably a

memory circuit which is addressed by the state of the cost bus 88. The change in cost data which is placed on the change in cost bus 90 and inputted to the adder 242 represents the change in cost which will occur if the proposed period or teacher change selected by the modulus memory 52 is accepted.

As units of the stored timetable may be moved or changed in blocks of units, units may be linked so that after change in cost data for an accessed unit is supplied to the adder 242, the control circuit 12 re-enters state 104 from state 112 via path 116. The previously determined change in cost data is temporarily stored, as described hereinafter, and submitted to the other input 244 of the adder 242 so as to be summed with the next change in cost data determined for the succeeding linked unit. The adder 242 therefore outputs total change in cost data for the linked units. A similar execution path for the teacher swap routine, wherein after the clash array is written to during state 118, corresponding to state 112, the next unit in the teacher change chain, according to the chain field 42, is accessed during state 120 and the control circuit 12 re-enters state 114.

The effect a period change of a class has on classes which clash with the current class of the accessed unit also needs to be examined in evaluating the total change in cost, and the class clash list circuit 54 has been included for this purpose, together with a subtractor 86 connected to the data lines of the Class From circuit 92 and the Class To circuit 94. If the current class of the accessed unit clashes with one or more other classes, as indicated by the flags field 41, during state 112 the class clash list circuit 54 uses the current class number from the output line 91 to address the clash list stored therein. Then, during a state 122, the circuit 54 outputs a clashing class number to the Class From and Class To circuits 92 and 94.

The class circuits 92 and 94 are addressed on the basis of the clashing class, instead of the current class, and the outputs of the circuits are passed to the two inputs of a subtractor 86. The total class cost for a given period is the sum of the products of the number of times the class occurs in the period and the number of times the clashing classes occurs in the period. For example, if C1 clashes with C2 and C3, then the cost will be #C1 * #C2 + #C1 * #C3, where # represents number of occurrences in a period. Therefore change in cost data for a clashing class is obtained by summing the differences between values representative of the number of times a clashing class has been allocated to the current period from the number of times the clashing class is allocated at the proposed period. These values are represented by the respective signals inputted to the subtractor 86, which during state 122 outputs change in cost data to the change in cost bus 90 for input to the adder 242 so as to update the total change in cost value. The next clashing class is read during state 124 following state 122. The operation is performed for each clashing class via a return path 128 until the clash list is exhausted, signified by a null class value (0). When completed, and no linked units exist, the state of the control circuit 12 proceeds to a decision point 130. A similar decision point 132 occurs in the teacher swap routine.

At decision points 130 and 132, the control circuit 12 examines the total cost change value outputted from a temporary store circuit 246 connected to the output of the adder 242. If the total cost change is less than zero, indicating a decrease in cost, the proposed change is accepted forthwith and operation of the control circuit 12 and apparatus 2 proceeds in accordance with an accept path 134. If the total cost change is greater than or equal to zero, the cost computation circuit 8 executes an annealing algorithm, at state 136, to determine whether to accept or

reject the proposed change outputted by the modulus memory 52. If the change is accepted, operation of the control circuit 12 proceeds in accordance with the accept path 134, otherwise operation proceeds in accordance with a reject path 136.

The annealing algorithm is an optimisation technique which is based on the theory associated with the cooling of a collection of vibrating atoms. When the atoms are at a high temperature they are free to move around and tend to move with random displacements. However, as an atomic mass cools, interparticle bonds force the atoms together. When the mass is cool, no movement is possible, and the configuration is set. If the mass is cooled quickly, the chance of obtaining an optimum solution, or configuration, is lower than if the mass is cooled slowly, or annealed. At any given temperature, a new configuration of atoms is accepted if the system energy for the mass is lowered. For annealing, if the energy is higher the configuration may also be accepted if the probability of such an energy increase is lower than that expected at the given temperature. The probability is given by $P(\blacktriangle E) = e^{-\blacktriangle E/kT}$, where k is Boltzmann's constant and ▲ denotes "change in" as normally represented by the Greek letter delta. The acceptance criteria is based on the physics of annealing.

Optimisation problems can be considered as having to configure or schedule a number of objects such that an objective function is minimized. By replacing the vibrating atoms with the objects, such as the elements of units in a timetable, and the system energy with the value of the objective function, such as the total cost of the timetable, the annealing acceptance criteria can be applied to optimisation problems. Applying the annealing algorithm to the timetable problem of the apparatus 2, an initial schedule is created by randomly scheduling the objects and storing them in the timetable memory circuit 4, as described

previously, and an initial temperature is set. Randomly suggested changes to the stored timetable are put forward and a total change in cost evaluated, which is effectively scaled by a temperature factor incorporated in an Exponential Lookup memory circuit 248, and the change accepted if the change in cost is negative, as discussed previously. If the change in cost, however, is greater than or equal to zero, the probability $P(\Delta c) = e^{-\Delta c/T}$, where $\Delta c/T$ is the scaled total change in cost, is evaluated. If the probability is greater than a randomly selected value outputted by the random number generator on line 60, the proposed change is accepted.

The temperature factor T is initially set so that initially all cost increases are accepted, and the temperature factor is gradually reduced by a predetermined cooling Rate factor, after predetermined number of swap routines are performed. The cooling rate is applied such that $T_n = T_{n-1} \times R$, where $0 = R < 1$, and T is a real number.

The advantage of the annealing algorithm over an algorithm which always seek a better solution (hill climbing algorithms), is that the annealing algorithm is less likely to direct a solution to a local minima, which may not be the optimum solution, simply because the annealing algorithm allows the cost to increase as well as decrease, whilst seeking a solution.

The probability $P(\Delta c)$ is evaluated in the Exponential Lookup memory circuit 248 of the Cost Computation circuit 8. The Exponential Lookup circuit 248 is addressed by the total change in cost outputted by the temporary store circuit 246 and outputs a signal representative of the probability to a comparator 250, which also receives the output of the random number generator 50. The comparator 250 of the Cost Computation circuit 8 outputs a signal, during state 136

which is received by the control circuit 12, on the basis of which the control circuit 12 proceeds to the accept path 134 or the reject path 138. The output signal is representative of whether the probability P(▲c) exceeds the random value outputted by the generator 50.

On the accept path 134, the apparatus 2 executes operations to write the accepted change into the timetable circuit 4 and to update the array circuits 70 accordingly. The first accessed unit used to calculate the total change in cost is first re-read during state 140 and the operations used to place change in cost data on the cost bus 88 are re-executed in state 142. However, this time, the state of the period crossbar switch 58 is reversed so the From circuits 92, 200, 204, 210, 214 receive the accepted change and increment the accessed location, in the same manner as the operation performed previously for the corresponding To circuits. The To circuits 94, 202,206, 212 and 216 receive the current period and decrement the assessed location, in accordance with the operation which was previously performed using the corresponding From circuit. The above operations are performed so as to adjust the corresponding From and To circuits so they contain the same data, in accordance with the accepted change.

A similar accept path 144 and reject path 146 are provided in the teacher swap routine, however, in this case, when the timetable memory circuit 4 is re-read in state 148 of the accept path 146, the teacher crossbar switch 56 state is reversed so as to swap the accessing teacher applied to the teacher circuits 214, 216, 220, 222, 224 and 226. Again, during a recompute cost state 150 in the accept path 144, the accessed records of the From circuits 214, 220 and 230 are incremented and the accessed contents of the To circuits 216, 222, and 226 are decremented to render the records of the corresponding From and To circuits the same.

The accept paths 134, and 144 are completed during states 152 and 154 where the increments and decrements are written to the appropriate array circuits 70 and the proposed change is written to the accessed unit of the timetable memory circuit 4.

The reject paths 138 and 146 include similar re-read states 156, 158, recompute cost states 160 and 162, and rewrite states 164 and 166 as the accept paths 134 and 144 and the same operations are performed as for the respective accept paths 134 and 144, except the states of the crossbar switches 56 and 58 are not altered. Therefore, during the states of the reject paths 138 and 146, the operations performed in the previous states 110 and 112, and 114 and 118, respectively are simply reversed as increment operations are performed instead of decrement operations, and vise versa, in the clash array circuit 6. Therefore, the clash circuits 70 are simply adjusted so as to return the circuits 70 to the status which existed before the swap routine began. During the rewrite states 164, 166 no alterations are made, of course, to the timetable memory circuit 4.

The accept and reject paths 134, 138, 144 and 146 are executed repetitively for all of the linked units used in generating the total cost change evaluated at the decision points 130, 132. When execution of the accept and reject paths has been completed, if the end of the timetable has not been reached for a period swap routine or the end of the teacher chain for a teacher swap routine has not been reached, then the next unit is accessed or the next unit in the teacher chain is accessed and operation proceeds to states 104 and 106, respectively. If the last record 16 in the timetable memory circuit 4 has been processed, or the end of the teacher chain has been reached, the control circuit 12 and the apparatus 2 enters a state 168 where a swap routine counter is decremented. The swap routine

counter is initially set to the number of routines which a user wishes the apparatus 2 to execute before halting. After the count has been decremented in state 168, the control circuit proceeds to the idle state 100 and then proceeds then to state 102 if the swap counter is not zero. If, of course, the swap counter is zero, the control circuit 12 and the apparatus 2 remain in the idle state 100, where the total cost of the timetable stored in the memory circuit 4 can be evaluated and displayed.

The operations performed by the apparatus 2 may be executed by a sophisticated software routine but as computer systems generally execute routines sequentially execution of the operations, most of which the apparatus 2 executes in parallel or simultaneously, involves a considerable amount of time. A commercially available personal computer would take approximately 2 to 3 weeks to execute the appropriate routines to produce an acceptable timetable solution. Instead, the apparatus 2 can be connected to the same personal computer via the pc interface 14 and using the apparatus 2, an acceptable solution can be obtained after a reasonable period of time.

A detailed circuit diagram of the apparatus 2 is provided in the accompanying Figures 3 to 9.

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as hereinbefore described with reference to the accompanying drawings.

DATED this 29th day of June, 1990.

COMMONWEALTH SCIENTIFIC AND INDUSTRIAL RESEARCH ORGANISATION
By its Patent Attorneys
DAVIES & COLLISON

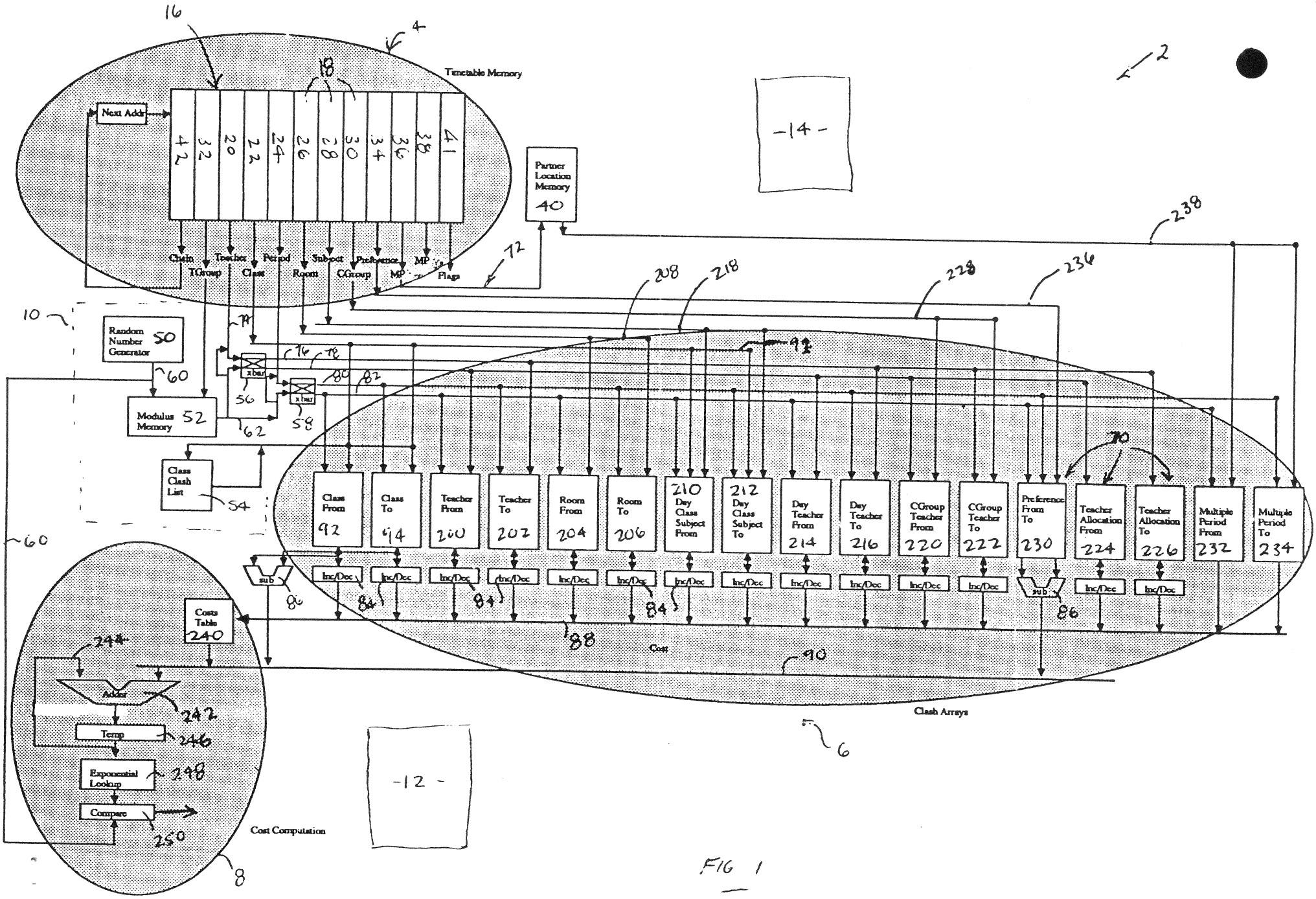
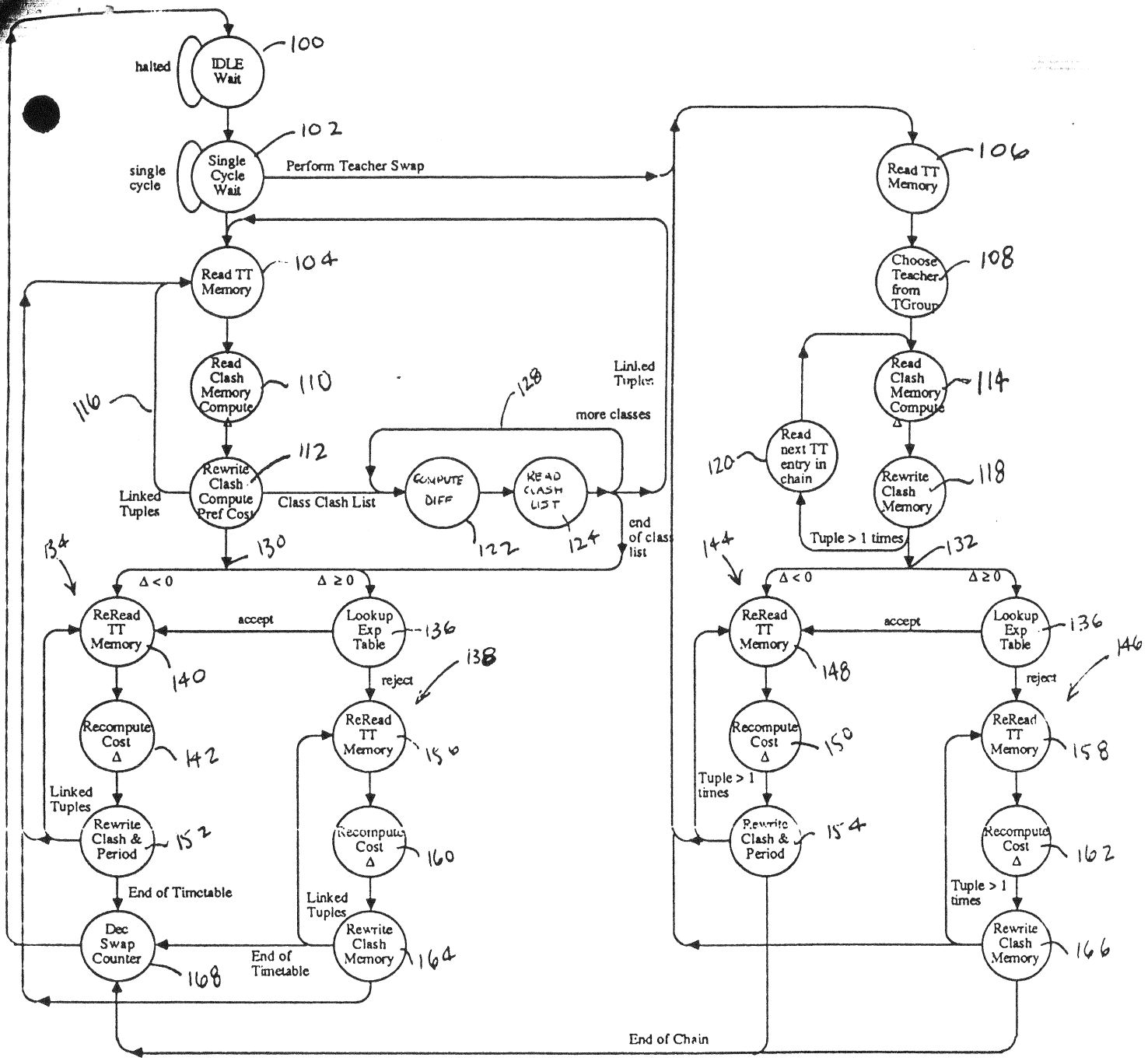
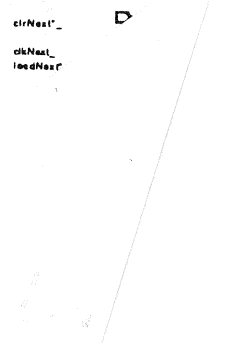

16
4
Timetable Memory
Next Addr
42
32
20
22
24
26
28
30
34
36
38
41
Chain
TGroup
Teacher
Class
Period
Room
Subject
CGroup
Preference
MP
MP
Flags
72
Partner Location Memory 40
-14-
2
238
208
218
228
236
10
Random Number Generator 50
60
xbar
56
xbar
58
Modulus Memory 52
62
Class Clash List 54
74
76
80
82
92
70
Class From 92
Class To 94
Teacher From 200
Teacher To 202
Room From 204
Room To 206
210 Day Class Subject From
212 Day Class Subject To
Day Teacher From 214
Day Teacher To 216
CGroup Teacher From 220
CGroup Teacher To 222
Preference From To 230
Teacher Allocation From 224
Teacher Allocation To 226
Multiple Period From 232
Multiple Period To 234
Inc/Dec
sub
84
86
88
Cost
90
Clash Arrays
6
60
Costs Table 240
244
Adder
242
Temp
246
Exponential Lookup
248
Compare
250
Cost Computation
8
-12-

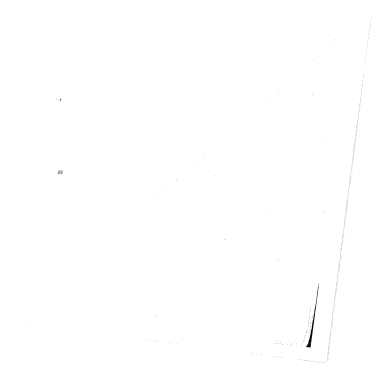
FIG 1

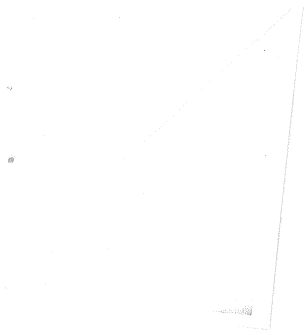


Note: This is a simplified state transition chart.
A number of optimisations have been omitted for clarity.

FIG. 2


clrNext"_
clkNext_
loadNext"





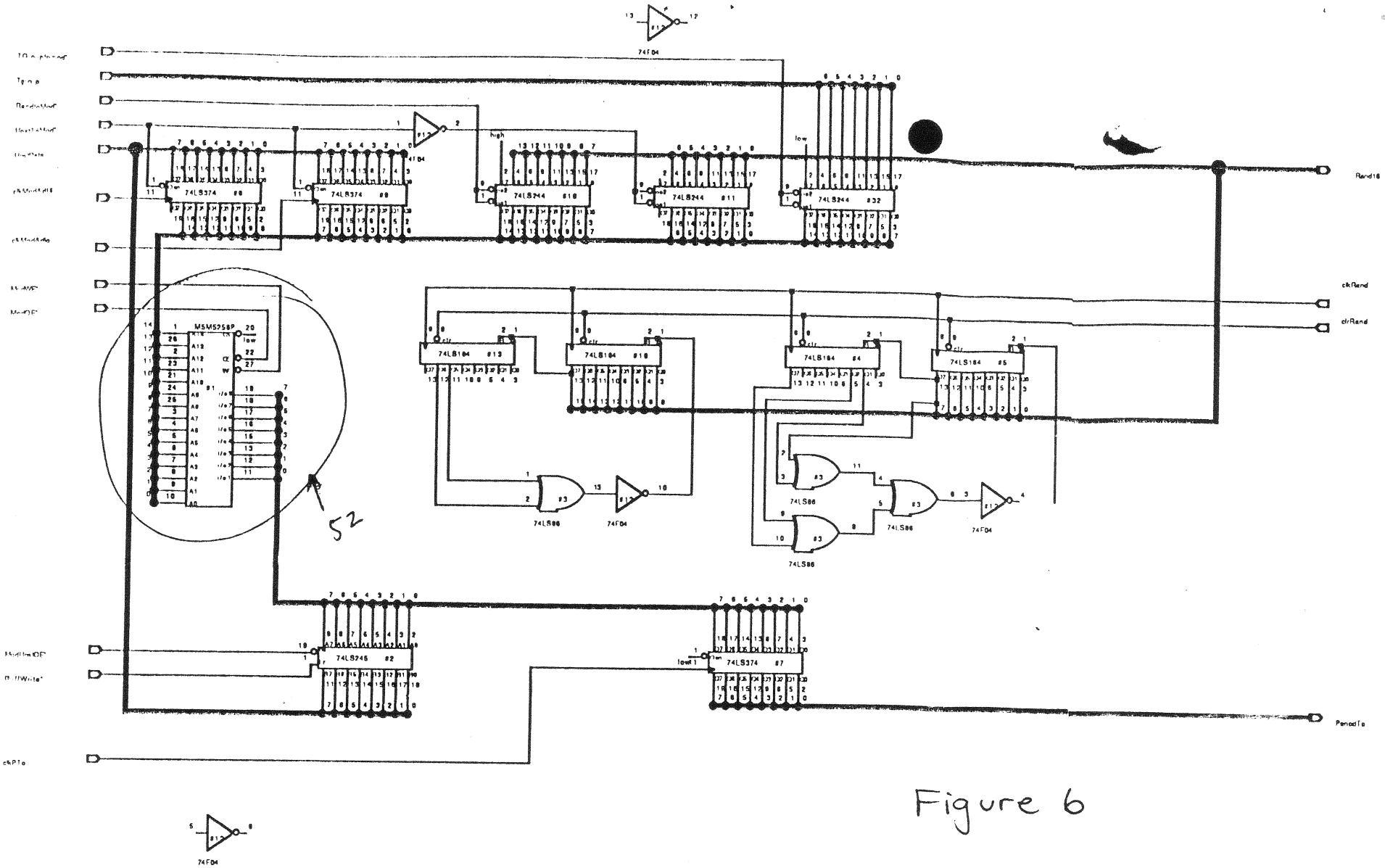
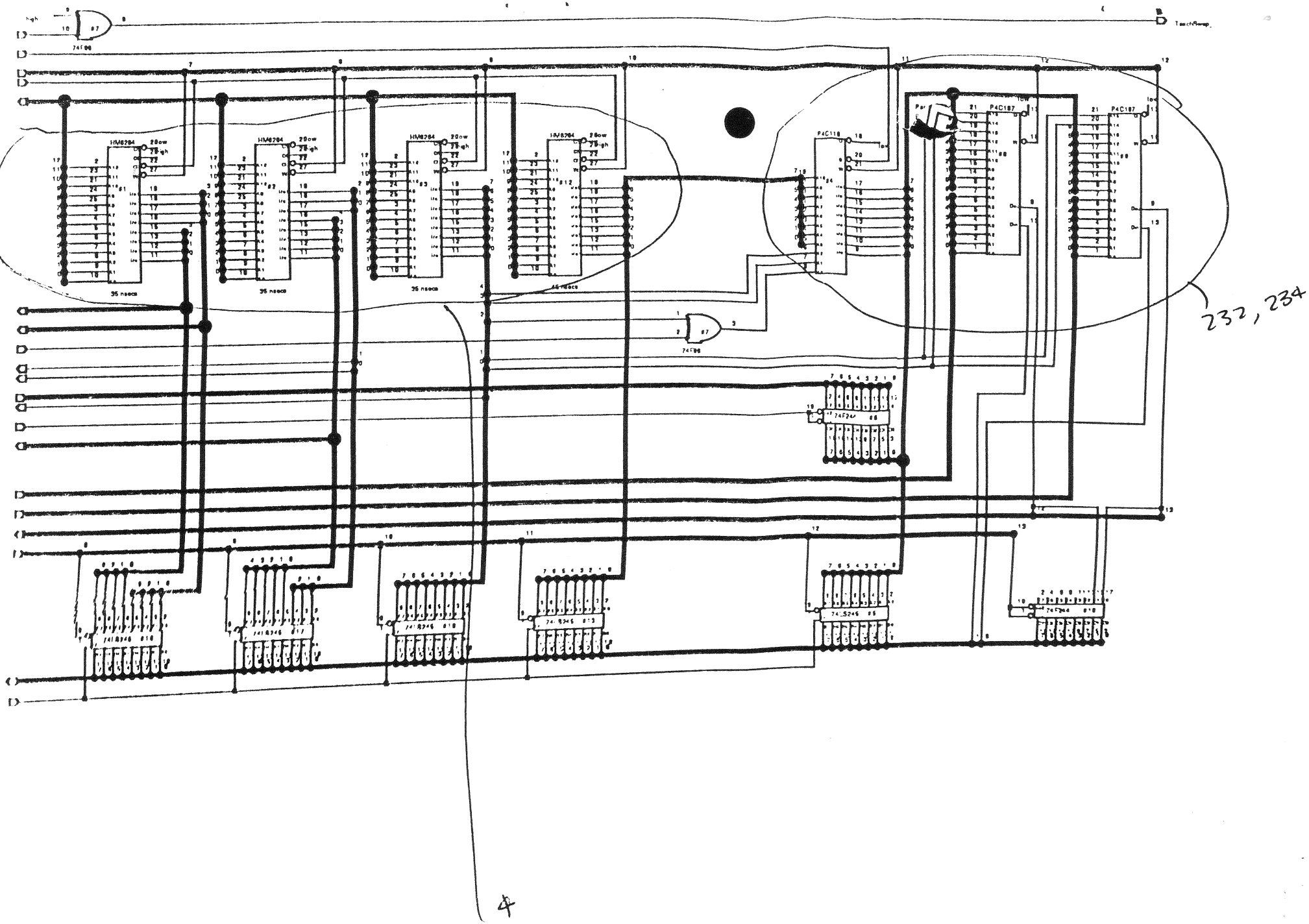


Figure 6

Random Number Generator	
V 2.0	Page 4
	McNET 187-507


232, 234
4