(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
1 May 2003 (01.05.2003)

PCT

(10) **International Publication Number**
WO 03/036468 A1

(51) **International Patent Classification⁷:** **G06F 9/38**

(21) **International Application Number:** PCT/SE01/02325

(22) **International Filing Date:** 24 October 2001 (24.10.2001)

(25) **Filing Language:** English

(26) **Publication Language:** English

(71) **Applicant** *(for all designated States except US)*: **TELEFONAKTIEBOLAGET LM ERICSSON (publ)** [SE/SE]; S-126 25 Stockholm (SE).

(72) **Inventor; and**
(75) **Inventor/Applicant** *(for US only)*: **LINNERMARK, Nils, Ola** [SE/SE]; Ullerudsbacken 74, 3 tr., S-123 73 Farsta (SE).

(74) **Agent: ERICSSON TELECOM AB**; Patent Unit Service and Backbone Networks, S-126 25 Stockholm (SE).

(81) **Designated States** *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

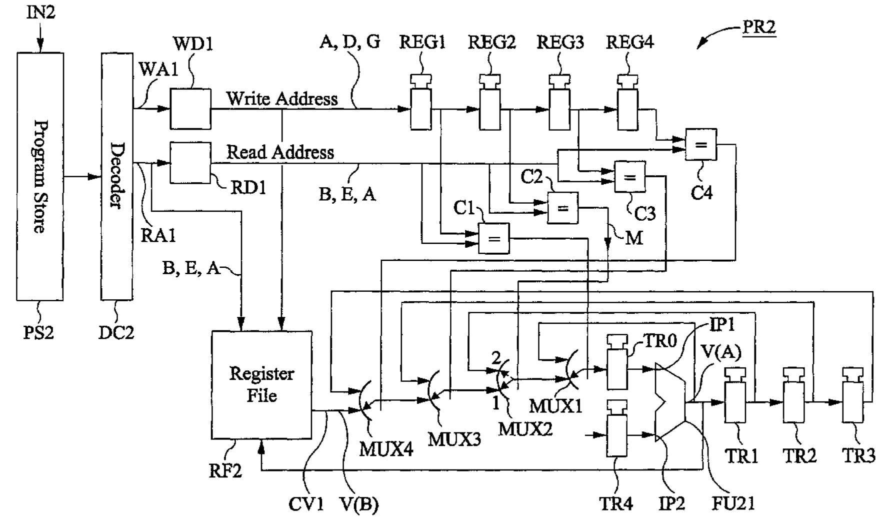
(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv)) for US only*

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** AN ARRANGEMENT AND A METHOD IN PROCESSOR TECHNOLOGY



(57) **Abstract:** A processor (PR2) has a functional unit (FU21) connected to series coupled temporary registers (TR21-TR23) and to a register file (RF2), which has an output connected to an input (IP1) of the functional unit via multiplexors (MUX1-MUX4). Read addresses (B, E, A) and write addresses (A, D, G) are sent to the register file and to a control means. The latter includes registers (REG1-REG4) and comparators (C1-C4) which control the multiplexors (MUX1-MUX4). On a read address (B) a value (V(B)) is sent to the functional unit (FU21) after the register file access time has lapsed. The functional unit performs an operation and the result (V(A)) is clocked through the temporary registers (TR1-TR3) and is sent to the register file (RF2). A later read address (A) coincides in the comparator (C2) with a write address (A) from the register (REG2), the multiplexer (MUX2) is switched and the result (V(A)) is fetched from the temporary register (TR1). The result (V(A)) can already be used, although it is under access in the register file (RF2) and can not yet be fetched from there.

WO 03/036468 A1

AN ARRANGEMENT AND A METHOD IN PROCESSOR TECHNOLOGY

TECHNICAL FIELD OF THE INVENTION

The present invention is related to an arrangement and a method in multiple-issue processor technology and more closely to an arrangement and a method to get a rapid and flexible multiple-issue processor.

DESCRIPTION OF RELATED ART

In processor design it is a desire to bring about a fast and flexible processor. In the processors, computation is performed in some type of device for computation and the results are stored in a register file. The results are fetched from the register file to be used in a subsequent computation of new results, which in turn can be stored in the register file. The process is controlled by a program in a program store. To make the processor more flexible and faster, reading and writing is performed for many computation devices simultaneously and independently of each other. A problem here is slow memories, e.g. the slow register file.

Multiple-issue processors allow multiple instructions to issue in a clock cycle. Commonly multiple-issue processors are divided up into two types, superscalar processors and VLIW (very long instruction word) processors. Superscalar processors issue varying numbers of instructions per clock cycle and can be either statically or dynamically scheduled, while VLIW processors issue a fixed number of instructions per clock.

The processor works at a certain clock frequency. As a general rule the performance increases with increasing clock frequency but there are also drawbacks to have a high clock frequency. One such drawback is that the pipeline length increases. Increasing pipeline length means that unpredictable or wrongly predicted jumps in the processor causes increasing delay, which means that the execution time increases. Another drawback is that high clock frequency design is generally difficult to implement. The clock distribution has to be done in such a way that minimal clock skew is inferred. To counteract this problem it is proposed to divide the design in different clock regions with substantial mutual clock skew, which affects the processor design.

Another factor that affects the processing speed is the propagation delay, which is made up of interconnect delays and gate delays. The interconnect delay is a continuously increasing part of the delay for each new technology generation. This means that the memory access will be more critical, since memory access time to large extent is interconnect delay.

The processing speed is affected by the memory design itself. Full custom design is performed on transistor level, the location of every transitor on a chip is optimized. There are many possibilities to optimize the processor design, and especially the memory design, for short delays. Making full custom design is anyhow costly and is not usable for small-size projects. An alternative to full custom design is cell library design, in which precompiled standard memories from a manufacturer are used. The cell libraries are placed on a chip in accordance with a specification from a customer. This design will give longer delays than full custom design but is cheaper. Still an alternative is gate array design, in which the standard cells are placed in a standard pattern on a chip by the manufacturer. Only the connection pattern can be designed by the customer. This design will give still longer delays.

Also another factor in the memory design affects the access delay. In both VLIW (very long instrucion word) and suoerscalar processor design multiported memories are used for the register file. The number of functional units can be high and every unit implies two read and one write port on the memory. The total number of ports is consequently high which will increase the access delay.

Renaming of register in the register file is a method used in out-of-order processors, that is processors that unlike VLIW processors execute the instructions in an order different from the instruction order in the code. In those processors the register data that is read at the operand-fetch stage is not always the correct data, since instructions not yet executed or speculatively executed can alter the register data. One method of implementing renaming is to store results from ALU (arithmetic logic unit) operations in temporary registers in the register file.

The U.S. patent No. 6,128,721 discloses a processor having an execution pipeline, a register file and a controller. The register file includes primary registers and temporary registers. It is mentioned that there are several problems with the introduction of temporary registers into the pipelines. In the patent the execution pipeline has a first stage for generating a first result and a second stage for generating a final result. The results are stored in the register file and the first result is made available if it is needed for an execution of a subsequent instruction. The lengt of the execution pipeline is reduced. The memory design for the register file and its access time is not discussed.

The international patent application with publication number WO 00/54144 discloses register file indexing in a VLIW processor to allow efficient implementation without the use of specialized vector processing hardware.

The U.S. patent No. 5.644.780 discloses a high speed register file for a VLIW or a superscalar processor.

SUMMARY OF THE INVENTION

The present invention is concerned with the main problem to get a rapid and flexible pipelined processor.

A further problem is to facilitate the use of a high processor clock frequency.

Another problem is to operate different processor computation devices independently of each other.

Still a problem is to facilitate the use of standard units in the processor design and manufacture and particularly, in an embodiment, using standard cell libraries including standard memories.

The problem is solved by storing computational results from the computation device in temporary registers, which are connected to respective of the computation device. The results are immediately available and can be utilized when required.

More closely the problem is solved by storing the computational result from a computation device in a set of temporary registers. The storing includes that the result is consecutively clocked through the set of registers and the result can be utilized when required. New results can be stored in this way one after the other. A time interval for the storing process can be selected by selecting the number of temporary registers. In an embodiment the time interval corresponds to the access time for a permanent memory device, i.e. it lasts until the computational result is stored in the permanent memory device, from which it then can be fetched when required.

A purpose with the invention is to get a rapid and flexible processor.

A further purpose is to derive advantage from high clock frequency in the processor.

Another purpose is to facilitate that different computation devices are operated independently of each other.

Still a purpose is to facilitate the use of standard units in the processor and particularly, in an embodiment, use of standard cell libraies including standard memory devices.

An advantage with the invention is that a processor with the temporary registers will be rapid and flexible.

A further advantage is that a high clock frequency can be fully utilized.

Another advantage is that different computation devices can be operated independently of each other.

Still an advantage is that standard units can be used in the processor, e.g. standard cell libraries including standard memories for a register file.

The invention will now be more closely described by prefered embodiments in connection with the enclosed drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram with an overview over a VLIW processor;

Figures 2a and 2b show block diagrams over alternative embodiments of parts of the processor;

Figure 3 shows a pipeline diagram for a processor;

Figure 4 is a block diagram showing more in detail logic circuits for the processor in figure 1;

Figure 5 is a block diagram over alternative logic circuits;

Figure 6 is a block diagram over still alternative logic circuits;

Figure 7 shows a block diagram with circuits for a superscalar processor; and

Figure 8 is a flow chart over a method in the processors in figures 1-6.

DETAILED DESCRIPTION OF EMBODIMENTS

Figure 1 is a block diagram showing an overview over a multiple-issue processor PR1. The processor has a program store PS1 with an input IN1 and with an output which is connected to a decoder DC1. It also has a first memory device in form of a register file RF1 for storing computational results and a second memory device in form of a data memory DM1. In an alternative a cache memory CM1 is connected to the data memory, as indicated by dotted lines. A first set of computation devices in form of functional units FU1, FU2,…FUM have inputs which are connected to the decoder and to outputs of the register file. Each of these functional units has an output, which is connected to a temporary register device in form of a pipeline tail of series coupled temporary registers. The functional unit FU1 is thus connected to the series coupled temporary registers TR11, TR12, TR13 and TR14, unit FU2 is coupled to temporary registers TR21, TR22, TR23 and TR24 and so on for the first set of functional units. A second set of functional units FU11 and FU12 have inputs which are connected to the decoder and to the data memory DM1. The functional units in the second set also have each a pipeline tail. The latter is rather long as the access time T2 for the data memory DM1 is rather long. In the figure is indicated that the functional unit FU11 has a pipeline tail of nine temporary registers TR111 to TR119. The processor PR1 works synchronously in wellknown manner and is controlled by clock pulses CL, which are indicated at some locations in the figure. The clock pulses are spread by a separate network, not shown in the figure.

The exemplified processor PR1 is a VLIW (very long instruction word) processor that works at a certain clock frequency, controlled by the clock pulses CL. The register file RF1 is of the previously mentioned type cell library and is rather slow with an access time T1. In the embodiment in figure 1 it takes five clock periods from the moment a value was received by the register file RF1 until the value has been stored and can be fetched. This delay is also the reason why there are four temporary registers in the pipeline tail, as will appear from the description below.

The functional units FU1, FU2,…FUM in the first set perform arithmetical and logical operations, e.g. the operation

$$R3=R1+R2 \qquad (1)$$

This operation is performed by the processor PR1 in the following manner. On an instruction I1 from the program store PS1 the functional unit FU1 fetches the values R1 and R2 from the register file RF1. The addition is performed and the result, the value R3, is sent to the register file RF1 to be stored there. The value R3 is also sent to the temporary register TR11 and is immediately stored there. All the operation is performed during a first clock period.

In a second clock period, directly following on the first, the program store PS1 sends an instruction I2 to the functional unit FU2 to perform an operation

$$R5=R3+R4 \qquad (2)$$

The functional unit FU2 fetches the value R4 from the register file RF1 and fetches the value R3 from the temporary register TR11. Note that the value R3 can not yet be fetched from the register file RF1, because its access time is so long and the value R3 is not yet stored there. The addition is performed and the result, the value R5, is sent to the register file RF1 to be stored and is also immediately stored in the temporary register TR21. The value R3 is clocked into the next termporary register TR12 in the pipeline tail during the second clock period. A new operation can be performed in the functional unit FU1 during the second clock period and a result is immediately stored in the temporary register TR11.

In a third clock period the program store PS1 sends an instruction I3 to the functional unit FU2 to perform the operation

R7=R6+R3 (3)

The value R6 is fetched from the register file RF1, the value R3 is fetched from the temporary register TR12, the addition is performed and the result, the value R7, is sent to the register file. It is also immediately stored in the temporary register TR 21. The earlier value R5 in the temporary register TR21 is clocked into the register TR22 and the earlier value R3 in the temporary register TR12 is clocked into the temporary register TR13.

In this manner the calculated values are successively clocked through the pipeline tails and can be fetched there until the pipeline tail ends. The value R3 for example can be fetched in a consecutive fifth clock period from the temporary register TR14. In a next clock period, a sixth period, it can be fetched from the register file RF1, because the value R3 is then stored there and can be fetced from there as rapidly as from one of the temporary registers.

The functional units FU11 and FU12 work together with their temporary registers and the data memory DM1 in the same way as decribed above for the functional units FU1-FUM.

The processor is flexible in that the different functional units can fetch values from each other´s temporary registers independently of each other. It is rapid in that a value calculated in one clock period can be used for computation already in the next clock period although the value is still under access in the register file. It is possible and efficient to use a high clock frequency although the register file can still be slow. A higher clock frequency results in that the access time lasts for more clock periods. Using a sufficiently long pipeline tail it is possible to use a calculated value immediately and during all the register file access time.

In figure 2a is shown an alternative to the pipeline tail for the functional unit FU1 in figure 1. The pipline tail having the temporary registers TR11, TR12… begins with a register TR10 in which a calculated value is always stored, also before it is sent to the register file RF1. In figure 2b is shown still an alternative with registers TR8 and TR9 at the inputs to the functional unit FU1.

In connection with figure 3 and figure 4 it will be more closely described how the functional unit with its pipeline tail is designed and how it works. The function will be descibed in connection with the following three calculations successively performed in one of the functional units:

$$\begin{aligned} &A = B + C \\ &D = E + F \qquad\qquad (4) \\ &G = A + H \end{aligned}$$

The letters A to H all denote adresses in different registers and corresponding values on these adresses will be denoted V(A), V(B) and so on in the description below.

Figure 3 shows pipeline diagrams, which together is an overview over how different jobs are pipelined in the processor. As an example it is shown how the above adresses B,E and A are clocked forward in the register file, having an access time of four clock periods. At a moment denoted by the clock CL=0 the address B is clocked into the register file. The register file will read the address B during the access time, denoted T1 in the figure. At next clock period CL=1 the address B is stepped forward and the next address E is clocked in. At clock period CL=2 the address A is clocked in. At a clock period CL=4 the address B is accessed and the value V(B) on the address B can be fetched from the register file.

Figure 4 shows a part of a single-issue processor PR2 having a functional unit FU21 with a pipeline tail of temporary registers TR1, TR2 and TR3 connected to its output. At one of its inputs IP1 the functional unit is connected to a temporary register TR0 and at the other input IP2 it is connected to a temporary register TR4. The processor has a program store PS2 which is connected to a decoder DC2. The decoder has two outputs, one write address otput WA1 and one read address output RA1. The write address output is connected to a first delay circuit WD1 including a number of registers and the read address output is connected to a second delay circuit RD1 also including a number of registers. The read address output RA1 is connected to a register file RF2, which has a certain access time of four clock periods and the delay circuits WD1 and RD1 have the same delay time, four clock periods. The first delay circuit WD1 is connected to the register file RF2 and to a set of series coupled registers REG1 to REG4. The second delay circuit RD1 is parallelly connected to a respective first input on a set of comparators C1 to C4. The comparators have each a second input which is connected to a respective one of the registers REG1 to REG4. The register file RF2 has an output CV1 which is connected to the the temporary register TR0 via a set of series coupled multiplexors MUX1 to MUX4. The multiplexors are connected to each other via each a first input and have each a second input which is connected to a respective one of the outputs from the functional unit FU21 and the temporary registers TR1, TR2 and TR3. The multiplexors have each a control input which is connected to an output on a respective one of the comparators C1 to C4. The output of the functional unit FU21 is connected to an input on the register file RF2.

In figure 4 the write addresses A, D and G and the read addresses B, E and A of the formula (4) are denoted.

The functional unit FU21 has a second input IP2 which is connected to a logic cicuitry which is of the same design as the above described logic, connected to the first input IP1. This logic circuitry is not shown, not to make the figure too complicated.

The function of the register pipeline tail TR1, TR2 and TR3 will be described below in connection with the processor PR2 in figure 4 and the formula (4). Some essential of the events during processing of the formula (4) will be denoted in Table 1 below to give an overview of the processing.

Table 1

CL1	CL2	CL3	CL4
A: REG1	D: REG1	G: REG1	
	A: REG2,C1	D: REG2,C1	
		A: REG3,C2	
B: C1-C4	E: C1-C4	A: C1-C4	
		MUX2 switched	
V(B): TR0	V(A)=V(B)+V(C):	V(A): TR2,TR0	V(G)=V(A)+V(H):
V(C): TR4	TR1,RF2	V(A): RF2	TR1,RF2
		V(H): TR4	V(A): RF2

In the table head four consecutive clock periods CL1-CL4 are given. For each clock period is then noted what happens in the registers REG1-REG4, after that what happens in the comparators C1-C4, then what happens whith the multiplexors and at last the calculations in the functional unit FU21 and the storing in the temporary registers TR0-TR3 and the register file RF2.

The processing of formula (4) begins with that the write addresses A, D and G are successively clocked from the decoder DC2 into the first delay circuit WD1. The read addresses B, E and A are successively clocked into the second delay circuit RD1 and these addresses are also successively clocked into the register file RF2. The read addresses C, F and H are clocked from the decoder, which is not shown in figure 4 or in table 1.

At a moment denoted as clock period CL1 the write address A is written into the register REG1, see upper left in the table. In the same clock period CL1 the read address B is sent to all the comparators C1-C4 and the value V(B) is sent from the register file RF2 and is stored in the register TR0. All these events take place during the clock period CL1 because the delay time of the delay circuits WD1 and RD1 are the same and correspond to the access time for the register file RF2. The value V(C) is written into the register TR4 but, as mentioned above, the cicuits for this writing are not shown in figure 4.

In the next clock period CL2 the write address D is written into the register REG1 and the write address A is written into the register REG2 and is sent to the comparator C1. The read address E is sent to all the comparators C1-C4. In the functional unit FU21 the value V(A)=V(B)+V(C) is calculated and the value V(A) is stored in the register TR1. The value V(A) is also sent to the register file RF2 to be stored there, which storing takes all the access time for the register file.

In the following clock period CL3 the write adress G is written into the register REG1, the write address D is written into the register REG2 and is sent to the comparator C1 and the write address A is written into the register REG3 and is sent to all the comparators C1-C4. The comparator C2 now has the address A on both its inputs and givs an output signal M to the multiplexor MUX2. This multiplexor switches from a position 1 to a position 2. The value V(A) is written into the temporary register TR2 and is also written into the temporary register TRO via the multiplexor MUX2. The value V(A) is also under storing in the register file RF2. In the same way as described, the value V(H) is written into the temporary register TR4.

Finally, in the clock period CL4, the value V(G)=V(A(+V(H) is calculated in the functional unit FU21 and is written into the temporary register TR1 and is also sent to the register file RF2 to be stored there. The value V(A), that was sent to the register file RF2 during the clock period CL2 is still under storing there.

In the description above, for simplicity, not all the events that take place during the processing of the formula (4) are mentioned. For example the write addresses G, A and D are stepped forward to the register REG4 and the value V(E) is calculated. The essential thing that appears is that the value V(A), calculated in the clock period CL2, can be utilized for calculation already in the clock period CL4, although it is still under storing in the register file RF2. In fact the value V(A) could have been utilized already in the clock period CL3, if required.

Figure 5 shows an alternative embodiment to the processor PR2 in figure 4. The processor in figure 5 has the program store PS2, the decoder DC2, the delay circuits WD1 and RD1, the registers REG1-REG4 and the comparators C1-C4. It also has the the register file RF2, the multiplexors MUX1-MUX4 and the temporary registers TR1-TR3. The difference is that the functional unit FU2 lacks the registers TR0 and TR4 at its inputs IP1 and IP2 but instead has a temporary register TR5 at its output. Values calculated in the functional unit FU2 are always stored in this register TR5 before they are stored in the register file RF2 or eventually returned to the input IP1.

Figure 6 shows still an alternative embodiment. In the figure the processor PR2 from figure 4 is shown within dotted lines. The processor PR2 is completed with a parallell functional unit FU41 having a pipeline tail of temporary registers TR41, TR42 and TR43. The embodiment in figure 6 is thus a multiple-issue processor. The pipeline tail TR41-TR43 is connected to locic circuit, in which a write address comes to a set of pipelined registers REG41, RFG42, REG43 and REG44, which are connected to a set of comparators C42, C43 and C44. The comparators are connected to a set of multiplexors MUX42, MUX43 and MUX44. As appears from the figure this parallell pipeline tail with its locic circuit is of the same design as corresponding elements in the processor PR2 and it also functions in the same manner. A dependency check in the processor PR2 can be done against all instructions corresponding to data in the parallell pipeline tail. In the embodiment it is assumed that the result from the functional unit FU41 will not be available in the functional unit FU21 until one clock period has passed to avoid a transportation delay that is added to the functional unit delay. The parallell functional unit FU41 with its pipeline tail of temporary registers TR41-TR43 and logical circuitry functions in the same way as the processor PR2. At a coincidence of the write and read addresses in e.g. the comparator C42 the multiplexor MUX42 is switched from a position 1 to a position 2. A value is then fetched from the temporary register TR41 and is transported to the temporary register TR0 at the input IP1 of the functional unit FU21.

Figure 7 shows a superscalar processor SCP1. Like the previously described processors it has a program store PS3 connected to a decoder DC3. The decoder is connected to a register file RF3 and to a delay circuit RD3, which is connected to a first set of comparators C71-C74 and to a second set of comparators C75-C77. The register file output is connected to a first set of multiplexors MUX71-MUX74 and to a second set of multiplexors MUX75-MUX77, which are connected to a computational unit COMP1 via a temporary register TR70. A first pipeline tail of temporary registers TR71-TR73 is connected to a first output of the computational unit and a second pipeline tail of temporary registers TR74-TR76 is connected to a second output of the computational unit COMP1. Outputs from the temporary registers are connected to the multiplexors, which are controlled by the comparators. The computational unit comprises a reservation stations block RS1, an execution block EX1 and a commit block CO1. A first address output from the commit block is connected to a first set of registers REG71-REG74 and to the register file RF3. A second address output from the commit block is connected to a second set of registers REG75-REG78 and to the register file RF3. Each of the comparators C71-C77 is connected to its respective one of the registers REG71-REG78. The reservation station RS1 fetches and buffers an operand as soon as it is available and when successive writes to a register appear, only the last one is used to update the register. When all operands actual for an instruction are available in the reservation station, the execution block EX1 executes the instruction. In the commit block then commit is made on the already executed instructions in a consecutive order, i.e. in the order they are read from the program store.

Figure 8 shows a flow chart for an overwiev over a method in connection with the above described processors. The method is also described in connection with the above Table 1. The method starts in a method step 80, in which values are stored in the memory device. In a next step 81 the write and read addresses are sent to the respective delay units, WD1 and RD1 or WD3 and RD3. The read addresses are also sent to the register file, RF1 or RF3, according to a step 83. The addresses are executed in the register file and when its access time is out the value on the read address is sent from the register file and the read and write addresses are sent from the delay units, see step 84. In a next step 85 calculations are performed in the functional unit FU21 or in the computational unit COMP1. The result of the calculations is stored in the first temporary register and is then successively clocked forward to the following temporary registers, see step 86. The storing in the register file begins according to a step 87. As the read and write addresses are clocked forward a coincidence of these addresses can occur in one of the comparison units, C1-C4 or C71-C74, according to a step 88. If this coincidence does not occure according to an alternative NO, new values are fetched from the register file in the step 84. When coincidence occure according to an alternative YES, a corresponding one of the multiplexors is switched. According to a step 89 a value from one of the temporary registers is fetched and is utilized in a calculation according to the step 85.

CLAIMS

1. A pipelined processor (PR1,PR2,SPC1) including:

- a memory device (RF1,DM1,CM1) for storing values (R1-R5) and having an access time (T1,T2); and

- at least one computational device (FU1,FU21) being connectable to the memory device and generating computational results (R3,R5,R7) that are stored in the memory device,

characterized in that the processor also includes:

- a temporary register device (TR11-TR14) connected to the computational device (FU1) and storing said computational results during at least a part of the access time (T1,T2) for the memory device (RF1); and

- a control means (REG1-REG4,C1-C4, MUX1-MUX4) connected to the temporary register device,

the control means being arranged to fetch the computational results (R3,R5,R7) from the temporary register device for use in further computations.

2. A pipelined processor (PR2,SPC1) including:

- a memory device (RF2,DM1,CM1) for storing values (V(B)) on addresses (B) and having a access time (T1,T2); and

- at least one computational device (FU1,FU21) for generating computational results (V(A)) in connection with address instructions, the computational device being connectable to the memory device,

characterized in that the processor also includes:

- a temporary register device (TR1-TR3) connected to an output of the computational device (FU21), the temporary register device storing said computational results during at least a part of the access time for the memory device; and

- a control means (REG1-REG4,C1-C4,MUX1-MUX4) connected to the temporary register device,

the control means being arranged to fetch the computational results (V(A)) from the temporary register device (TR1-TR3) on receiving corresponding address instructions (A), the results being intended for use in further computations.

3. The processor according to claim 2, characterized in that the control means (REG1-REG4,C1-C4,MUX1-MUX4) is arranged, when fetching said computational results (VA)), to compare a read address (A) with a write address (A) and, on coincidence of the addresses, to fetch the corresponding computational result (V(A)) from the temporary register device (TR1-TR3).

4. The processor according to any of claims 1-3, characterized in that the computational results (V(A)) are used in the further computations during the memory device (RF2) access time.

5. The processor according to any of claims 1-4, characterized in that the temporary register device includes a pipeline tail (TR11-TR14,TR1-TR3) of series coupled temporary registers.

6. The processor of claim 5, characterized in that said pipeline tail (TR11-TR14,TR1-TR3) includes at least three temporary registers.

7. The processor according to any of claims 1-6, characterized in that the memory device (RF1,RF2) is a register file.

8. The processor according to any of claims 1-6, characterized in that the memory device (CM1) is a first level data cache memory.

9. The processor according to any of claims 1-8, characterized in that the processor ((PR1) is a multiple-issue processor.

10. The processor according to any of claims 1-8, characterized in that the processor (PR2) is a single-issue processor.

11. The processor according to any of claims 1-9, characterized in that the processor (PR1,PR2) is a VLIW processor.

12. The processor according to any of claims 1-9, characterized in that the processor (SPC1) is a superscalar processor.

13. A method in a pipelined processor (PR1,PR2,SPC1), the processor including:

- a memory device (RF1,DM1,CM1); and
- at least one computational device (FU1,FU21,COMP1),

the method including:

- storing (80)values (V(B)) in the memory device, the memory device having an access time (T1,T2); and
- generating (85) computational results (V(A)) in the computational device (FU1,FU21,COMP1),

characterized in that the method also includes:

- storing (86) said computational results in a temporary register device (TR11-TR14) during at least a part of the access time for the memory device;

- controlling (88) the temporary register device by a control means (REG1-REG4,C1-C4,MUX1-MUX4); and

- fetching (89) the computational results (V(A)) from the temporary register device (TR11-TR14) by the control means for use in further computations (85).

14. A method in a pipelined processor (PR1,PR2,SPC1), the processor including:

- a memory device (RF1,DM1,CM1); and

- at least one computational device (FU1,FU21,COMP1),

the method including:

- storing (80) values (V(B)) on addresses (B) in the memory device, the memory device having an access time /T1,T2); and

- generating (85) computational results (V(A) in the computational device in connection with address instructions (A),

characterized in that the method also includes:

- storing (86) said computational results (V(A)) in a temporary register device (TR11-TR14) during at least a part of the access time (T1,T2) for the memory device;

- controlling (88) the temporary register device by a control means (REG1-REG4,C1-C4,MUX1-MUX4); and

- fetching (89) the computational results (V(A)) from the temporary register device (TR11-TR14) by the control means for use in further computations (85).

15. A method according to claim 14 characterized in

- comparing (88) in the control means a read address and a write address;

- noting (YES) a coincidence of the addresses; and

- fetching (89) the corresponding computational result (V(A)) from the temporary register device (TR11-TR14) for further computations (85).

16. The method in the processor according to any of claims 13-15, characterized in storing (86) the computational results (V(A)) in the temporary register device (TR11-TR14) during all the access time (T1,T2) for the memory device.

17. The method in the processor according to any of claims 13-16, characterized in:

- storing (86) the computational result (V(A)) in a first one (TR11) of at least two series coupled temporary registers (TR11,TR12,TR13) of the temporary register device during a processor clock period (CL); and

- clocking successively the computational result (V(A)) through the series coupled temporary registers (TR11-TR14).

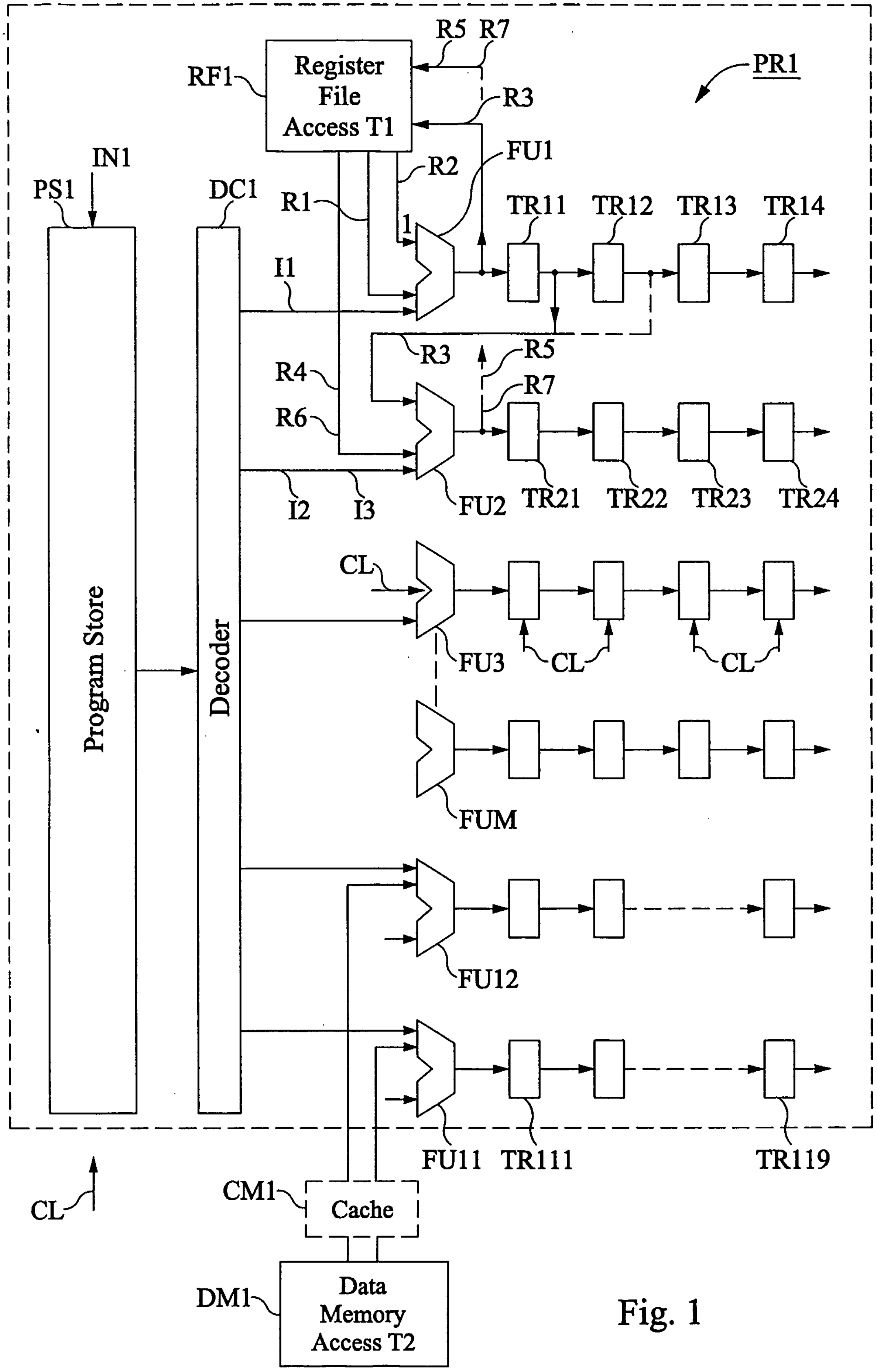

R5 R7
Register File Access T1
RF1
PR1
R3
FU1
R2
IN1
PS1
DC1
R1
1
TR11
TR12
TR13
TR14
I1
R3
R5
R4
R7
R6
FU2
TR21
TR22
TR23
TR24
I2
I3
CL
Program Store
Decoder
FU3
CL
CL
FUM
FU12
FU11
TR111
TR119
CL
CM1
Cache
DM1
Data Memory Access T2

Fig. 1

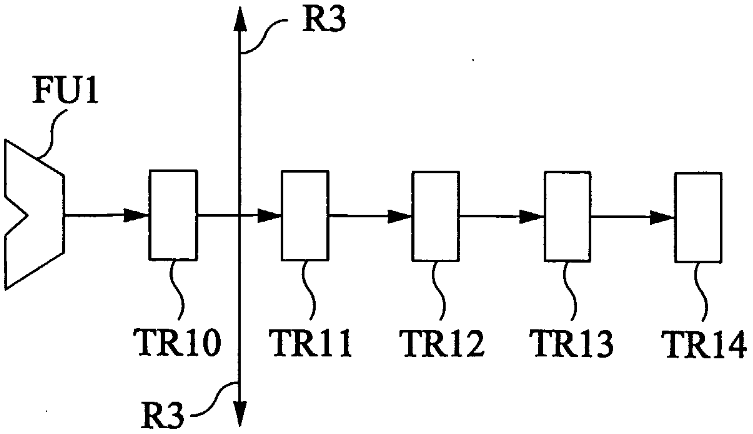


Fig. 2a

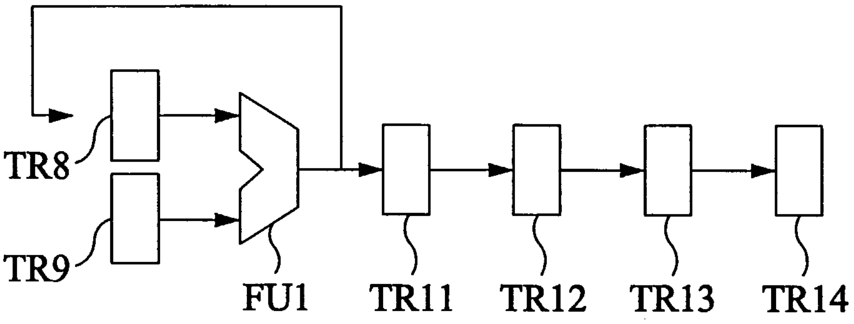


Fig. 2b

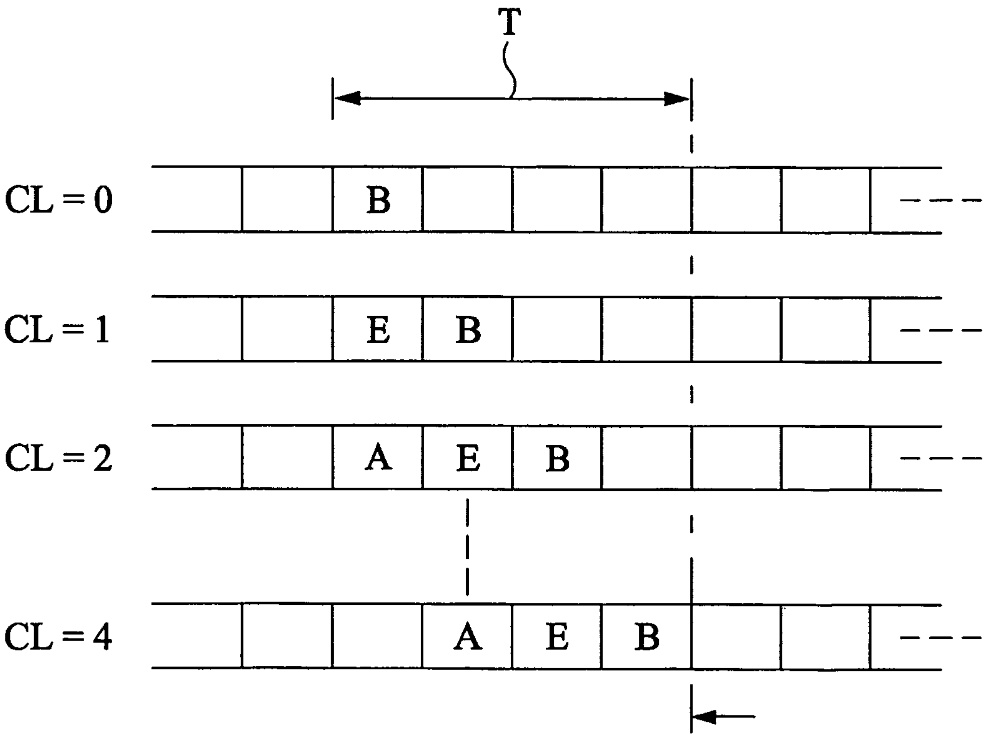


Fig. 3

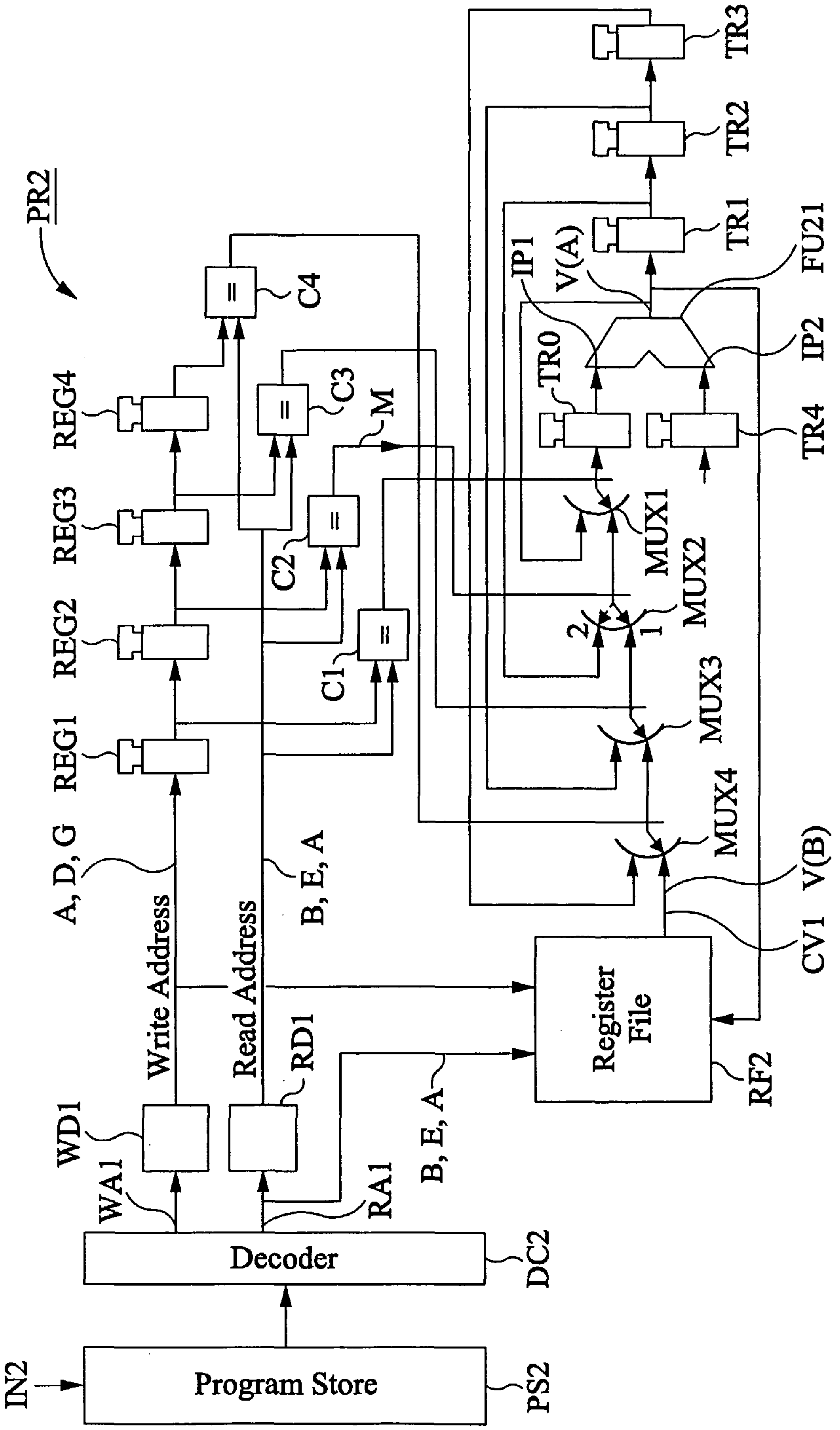

PR2
REG1
REG2
REG3
REG4
C1
C2
C3
C4
M
TR0
TR1
TR2
TR3
TR4
IP1
IP2
V(A)
FU21
MUX1
MUX2
MUX3
MUX4
2
1
A, D, G
B, E, A
Write Address
Read Address
WD1
RD1
WA1
RA1
B, E, A
Register File
RF2
CV1
V(B)
Decoder
DC2
Program Store
PS2
IN2

Fig. 4

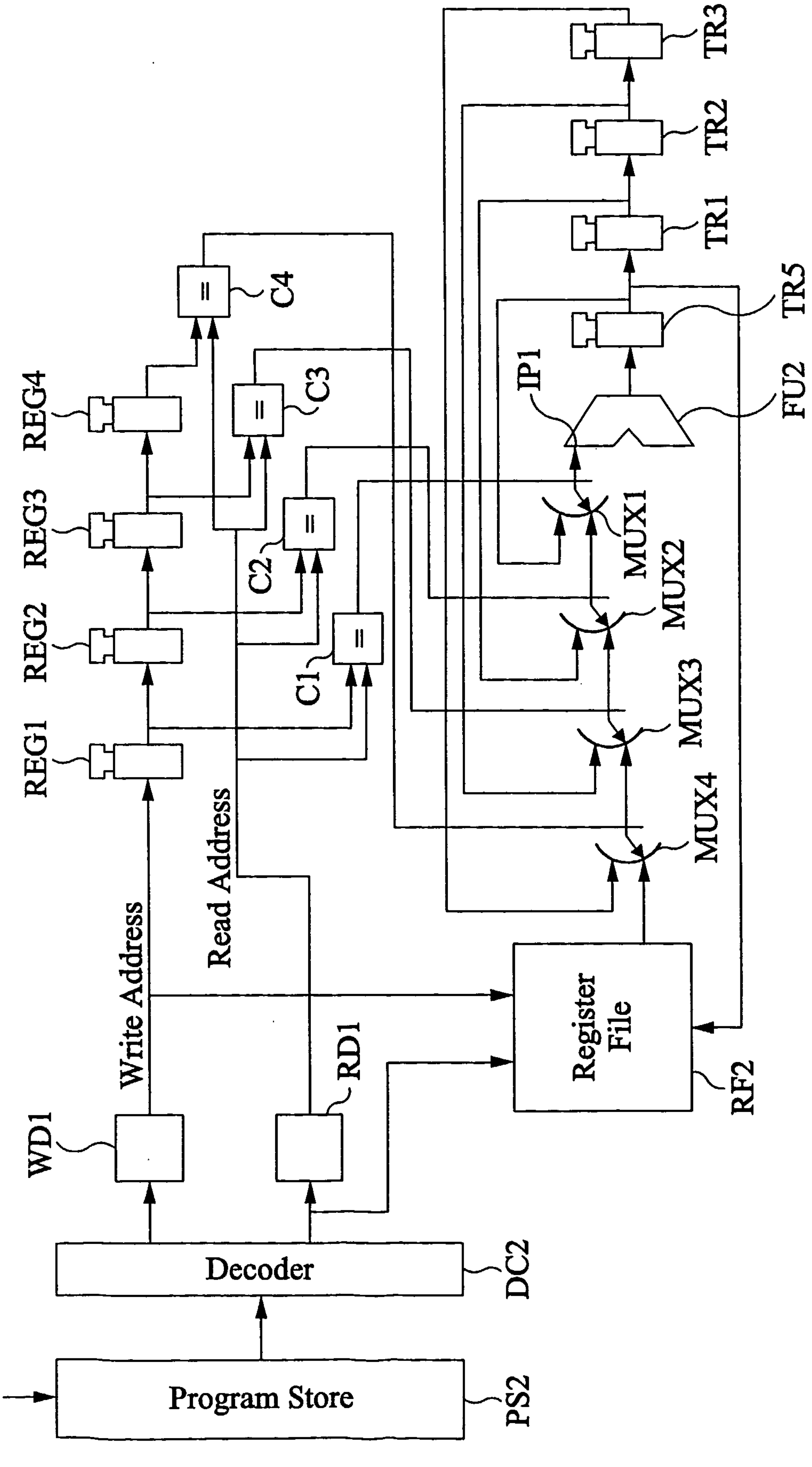

Program Store
PS2
Decoder
DC2
WD1
RD1
Write Address
Read Address
Register File
RF2
REG1
REG2
REG3
REG4
C1
C2
C3
C4
=
MUX1
MUX2
MUX3
MUX4
IP1
FU2
TR5
TR1
TR2
TR3

Fig. 5

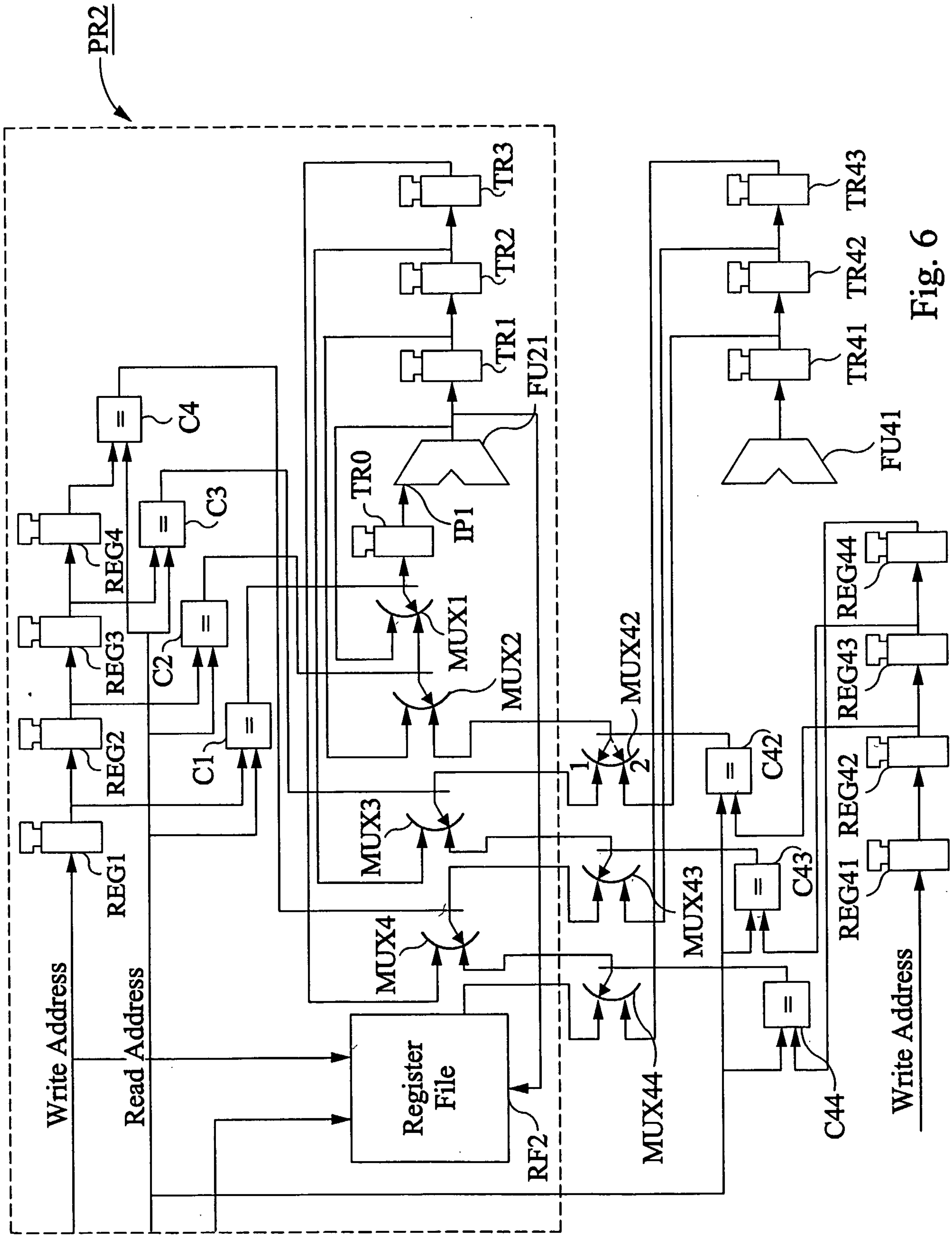

PR2
Write Address
Read Address
REG1
REG2
REG3
REG4
C1
C2
C3
C4
=
TR0
IP1
MUX1
MUX2
MUX3
MUX4
FU21
TR1
TR2
TR3
Register File
RF2
MUX42
1
2
MUX43
MUX44
C42
C43
C44
TR41
TR42
TR43
FU41
REG41
REG42
REG43
REG44
Write Address

Fig. 6

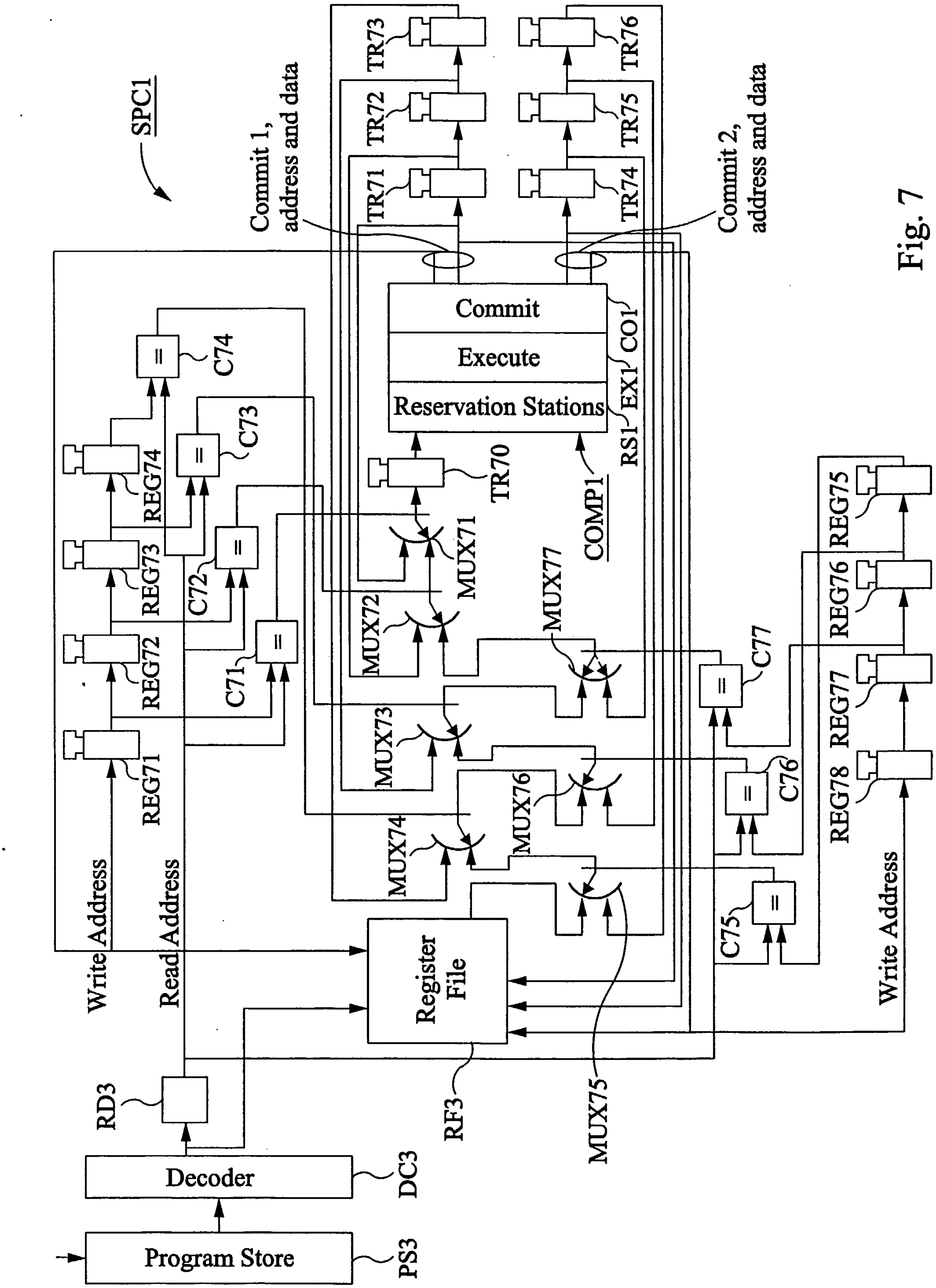

SPC1
Commit 1, address and data
Commit 2, address and data
TR71
TR72
TR73
TR74
TR75
TR76
TR70
Commit
Execute
Reservation Stations
CO1
EX1
RS1
COMP1
C71
C72
C73
C74
C75
C76
C77
=
REG71
REG72
REG73
REG74
REG75
REG76
REG77
REG78
MUX71
MUX72
MUX73
MUX74
MUX75
MUX76
MUX77
Write Address
Read Address
Write Address
Register File
RF3
RD3
Decoder
DC3
Program Store
PS3

Fig. 7

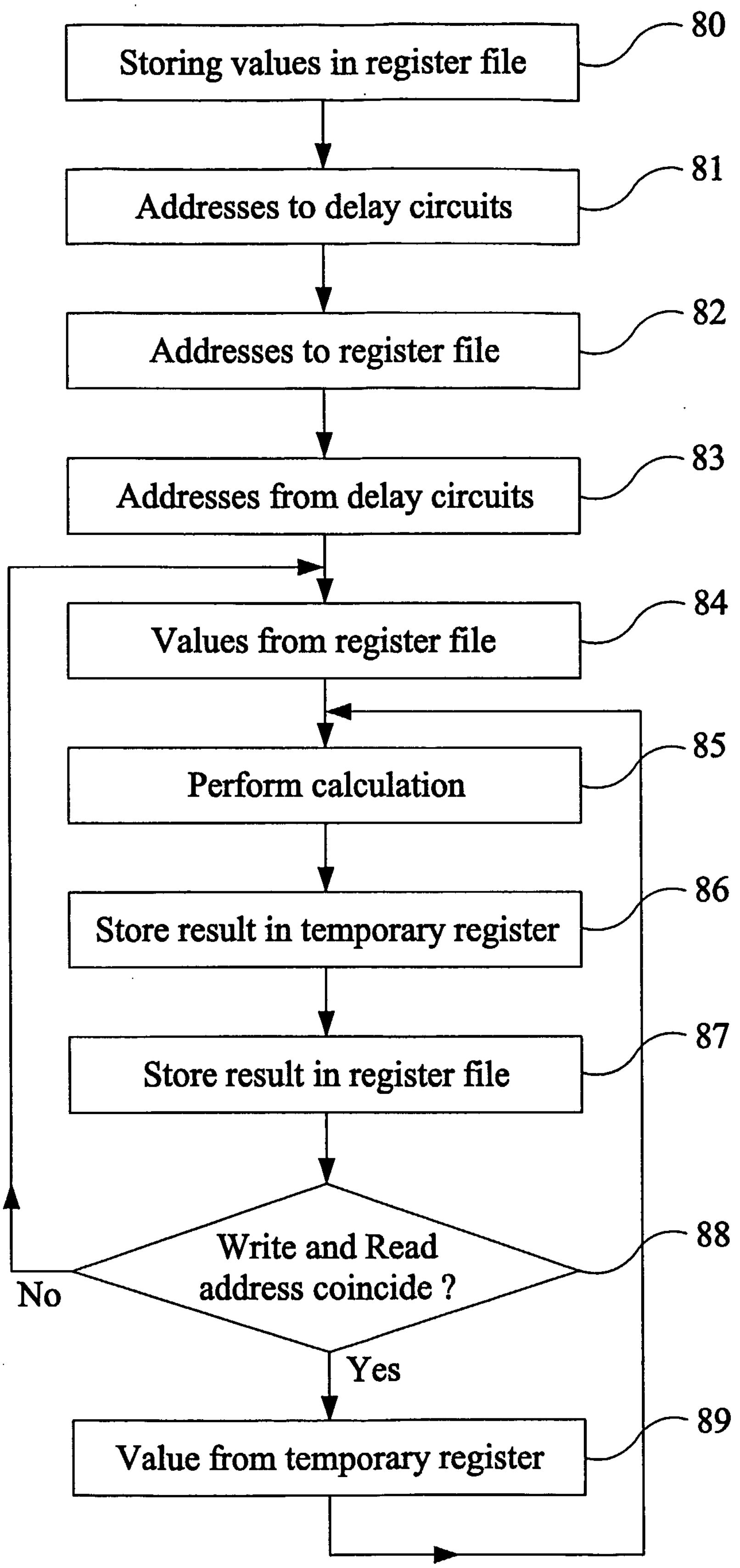

Storing values in register file
80
Addresses to delay circuits
81
Addresses to register file
82
Addresses from delay circuits
83
Values from register file
84
Perform calculation
85
Store result in temporary register
86
Store result in register file
87
Write and Read address coincide ?
88
No
Yes
Value from temporary register
89

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 01/02325

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 9/38

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5117493 A (JENSEN, E.H.), 26 May 1992 (26.05.92), figure 3, abstract --	1-17
X	US 5964862 A (LEUNG, A.T. ET AL), 12 October 1999 (12.10.99), column 1, line 65 - column 3, line 60, figure 1, abstract --	1-17
X	EP 0898226 A2 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.), 24 February 1999 (24.02.99), figure 1, abstract --	1-4,7-16
A	US 6233670 B1 (IKENAGA, C. ET AL), 15 May 2001 (15.05.01), figure 2, abstract --	1-17

□ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 10 June 2002	Date of mailing of the international search report 13 -06- 2002
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. + 46 8 666 02 86	Authorized officer Jenny Forss /OGU Telephone No. + 46 8 782 25 00

INTERNATIONAL SEARCH REPORT

Information on patent family members

01/05/02

International application No.
PCT/SE 01/02325

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5117493 A	26/05/92	GB 2234833 A,B	13/02/91
		GB 9004698 D	00/00/00
		KR 9711208 B	08/07/97
US 5964862 A	12/10/99	NONE	
EP 0898226 A2	24/02/99	CN 1208894 A	24/02/99
		JP 2869414 B	10/03/99
		JP 11126154 A	11/05/99
		US 6260136 B	10/07/01
US 6233670 B1	15/05/01	DE 4207148 A	24/12/92
		JP 4367936 A	21/12/92
		US 5636353 A	03/06/97