Stage 6: Support for all DT instructions features

The features to be supported include byte and half word transfers (signed and unsigned), auto increment/decrement with option of pre/post indexing.

Word, Half-word and Byte transfers

We need to introduce a combination circuit between the processor and memory which does the required transformation of words into half-words / bytes and vice versa. Let us call it PMconnect.

Let Adr[31-0] denote the address for memory. This is a byte address. Out of the 32 bits of address, 30 bits Adr[31-2] specify the address of a word and 2 bits Adr[1-0] specify a byte within that word or Adr[1] specifies a half-word within that word.

To make it possible to write a byte or a half-word in the memory, one solution is to have 4 write enable signals for memory, one for each byte. Let these be denoted by MW[3-0]. MW[0] is for the least significant byte and MW[3] is for the most significant byte. The memory may be addressed by Adr[31-2] to select a word (for load as well as for store). For store instructions, MW controls which byte(s) get written. For load instructions, we can read an entire word and transfer selective portion to the register file.

Let Rout[31-0] denote the source data for store instructions (i.e., contents of RF[IR[15-12]]) and Rin[31-0] denote the data to be put in register file for load instructions (i.e., data destined for RF[IR[15-12]].

Let Min[31-0] denote the data input of memory and Mout[31-0] denote the data output from memory.

Inputs and outputs of PM connect are shown in the adjoining figure. This module does not need to look at the instruction in totality. It needs to only know which is the instruction class and if the class is DT, which DT instruction it is. As the delay of this circuit is not large, no additional control state and no additional registers are required to be introduced.

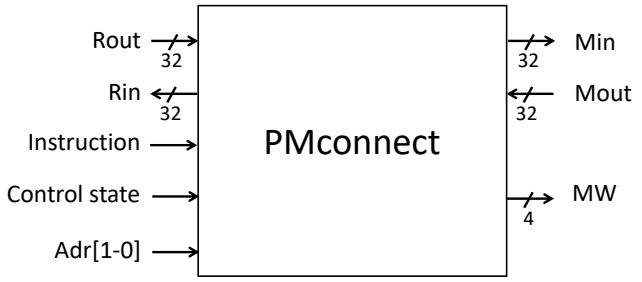

Table I shows how Min and MW are determined from Rout, Instruction and Adr[1-0]. MW output shown is for the control state corresponding to memory write. In other states MW = 0000. Table II shows how Rin is obtained from Mout, Instruction and Adr[1-0].

Table I: Determining Min and MW

Instruction	Adr[1-0]	Min[31-24]	Min[23-16]	Min[15-8]	Min[7-0]	MW[3-0]
STR	00	Rout[31-24]	Rout[23-16]	Rout[15-8]	Rout[7-0]	1111
STRH	00	Rout[15-8]	Rout[7-0]	Rout[15-8]	Rout[7-0]	0011
	10					1100
STRB	00	Rout[7-0]	Rout[7-0]	Rout[7-0]	Rout[7-0]	0001
	01					0010
	10					0100
	11					1000

Table II: Determining Rin

Instruction	Adr[1-0]	Rin[31-24]	Rin[23-16]	Rin[15-8]	Rin[7-0]
LDR	00	Mout[31-24]	Mout[23-16]	Mout[15-8]	Mout[7-0]
LDRH	00	00000000	00000000	Mout[15-8]	Mout[7-0]
	10			Mout[31-24]	Mout[23-16]
LDRSH	00	bbbbbbbb	bbbbbbbb	Mout[15-8]	Mout[7-0]
	10	dddddddd	dddddddd	Mout[31-24]	Mout[23-16]
LDRB	00	00000000	00000000	00000000	Mout[7-0]
	01				Mout[15-8]
	10				Mout[23-16]
	11				Mout[31-24]
LDRSB	00	aaaaaaaa	aaaaaaaa	aaaaaaaa	Mout[7-0]
	01	bbbbbbbb	bbbbbbbb	bbbbbbbb	Mout[15-8]
	10	cccccccc	cccccccc	cccccccc	Mout[23-16]
	11	dddddddd	dddddddd	dddddddd	Mout[31-24]

Here **a** = Mout[7], **b** = Mout[15], **c** = Mout[23] and **d** = Mout[31]

Implementing Auto-indexing

Auto-indexing requires the address calculated by adding offset to the base be written back into the base register. In case of pre-indexing, memory is accessed using the computed address and in case of post-indexing, memory is accessed with just the base address.

The additional operation required for auto-indexing is the write back of computed address into the base register. We need to decide in which cycle the base register is written back - whether it can be done in an existing control state or a new control state needs to be introduced. Load instructions already have a cycle in which the data read from memory is written into register file. The cycle for base register write back has to be a cycle different from this cycle because only a single write port is available with register file. However, it can be the same cycle in which memory is accessed.