X3. Instruction set(s) and instruction times. Give the original notation, together with explanations and annotations that can be understood by a modern reader. Where appropriate, give the memory map and any addressing/indexing/input-output conventions which machine-level programmers would need to know about. Give accurate references to the sources of all information – see X5 below. (Contributes to database blocks D2, D5).

E5X3. Elliott 900 Series: Instruction Set and Times.

18-bit machines.

920A.

Very similar to the 920B described in detail below, but with the following limitations, deduced mainly from "Programming Compatibility of 920 Series Computers" [2]:

Function 8	Unconditional Jump does affect the Q-register.
Function 11	Store SCR affects the Q-register but not in a useful way.
Function 14	Does not provide any Block Transfer facilities.
Function 15	Paper Tape Input can only read 7 tracks of the tape.

920B, 903, ARCH 9000.

The sections "Register Structure", "Order Code" and "Instruction Modification" below are taken from "Elliott MCS 920B Computer Programming" [4]. The sections "Priority Level Program Organisation", "Initial Instructions" and "Store Addressing" below are taken from the "Elliott MCS 920B Facts" card [5] and "Elliott 903 Computer Facts" cards [6 & 7]. *Comments in italic are not part of those original texts.*

Register Structure.

The computer carries out its various functions through an array of registers, an Adder unit and a Collate unit. The registers and their roles (see schematic below) are:

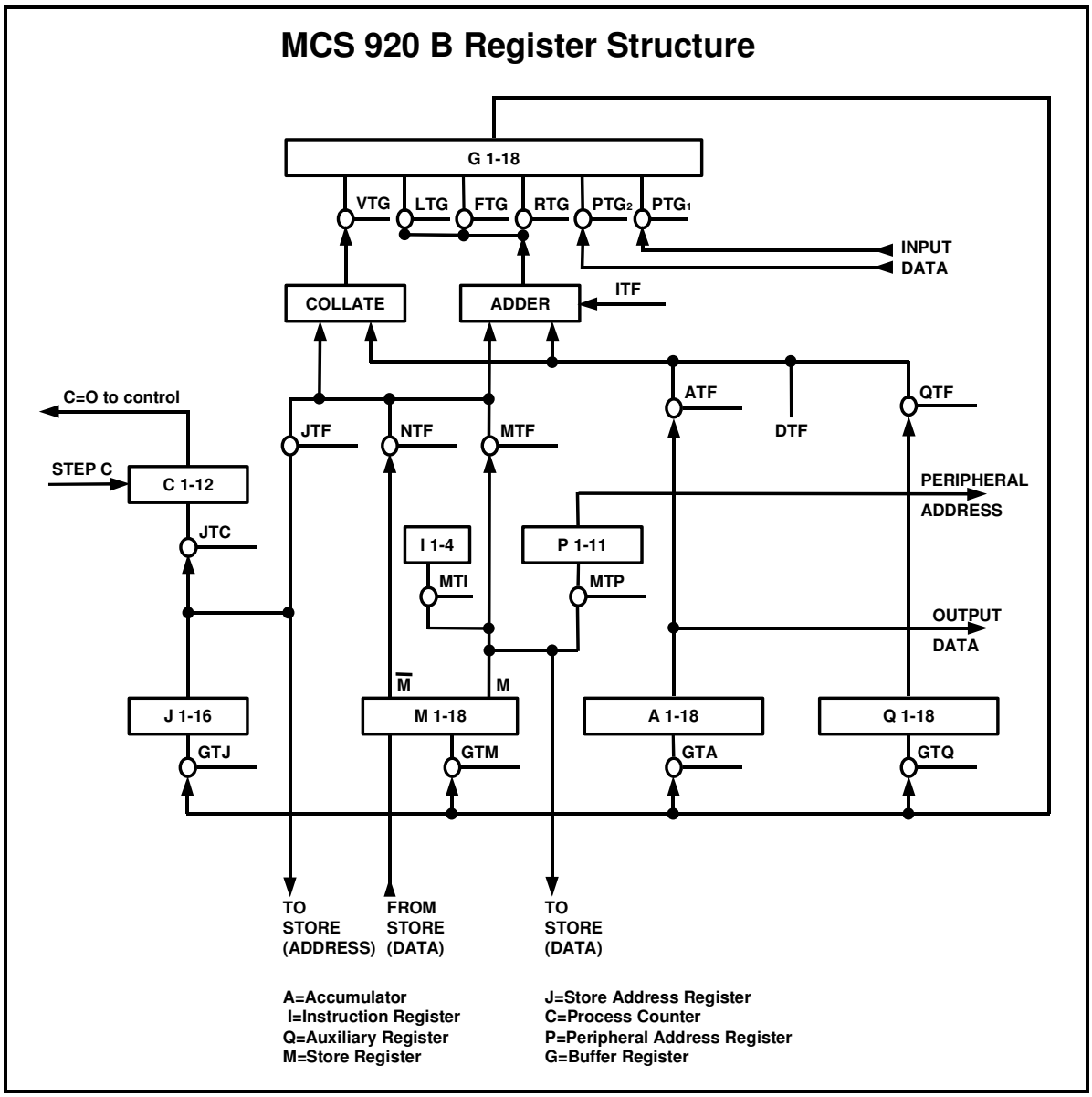
The J. Register of 16 binary digits is used to hold numbers specifying store locations.

The M. Register of 18 binary digits is used as a buffer register between the store and the computer.

The I. Register of 4 digits holds the *function* number specifying the current instruction.

The A. Register is an accumulator with a capacity of 18 digits which is used to hold the result of calculations prior to transfer to the store or output channel.

The Q. Register of 18 digits is used to hold information temporarily during a computation and also as an extension of the accumulator *to the right* for numbers containing more than 18 significant digits.



Of these registers, the only ones which convey information from one instruction to the next, and so are of interest to the programmer, are the A-Register (or Accumulator) and the Q-Register (or Auxiliary Register).

The SCR (Sequence Control Register) and B-Register (Modifier) are also of interest to the programmer, but do not feature on this schematic because they are held in the store.

The Adder unit is an 18 digit parallel adder. Total Carry propagation time is 1 μsec.

Input (a) may be selected from
- (i) Register M contents
- or (ii) Inverse of register M contents
- or (iii) Register J contents

Input (b) may be selected from
- (i) Register A contents
- or (ii) Register Q contents
- or (iii) -2^{-17}
- or (iv) 8192×2^{-17} (This enables the computer to collate out the address part of an instruction).

The collate unit generates the logical product of the (a) and (b) inputs to the adder. The outputs from the adder and collate unit may be routed to any one of the registers A, Q, M, J. The output from the adder may be shifted left *via LTG* or right *via RTG,* by one binary digit position before transfer to the destination register.

The Peripheral Address Register specifies the peripheral unit selected in Block Transfer and Input/Output instructions.

The Process Counter is used to count the steps in multiplication, division, and shifting operations. It is also used to count words during Block Transfer functions.

Order Code.

The 18 digit binary word may represent either a number or an instruction. Positive numbers are represented directly: negative numbers by their complement with respect to 2. The range of numbers that may be represented is from -1 to $+1-2^{-17}$. When the word is an instruction the format is:

B	**F**	**N**
Digit 18	Digits 17–14	Digits 13–1
B is a modifier	F is a function	N is an address

The 23 instructions are given below. It should be noted that the Sequence Control Register is incremented before the current instruction is obeyed.

The Auxiliary Register fulfils various roles. It should be noted that B modification affects its contents as do instructions 0, 2, 7, 9, 11 and 13.

Function Number	**Title**	**Specification**
0	Set B register	Place in the B register of the current program level, and in the Q register, the contents of the store location specified by N. The content of A is not affected by this function.
1	Add	Add the contents of the store location specified by N to the accumulator. The contents of Q are not affected by this function.

Function Number	Title	Specification
2	Negate and Add	Negate the contents of the accumulator and add the contents of the store location specified by N. The contents of N are also placed in the Q register by this function.
3	Store Auxiliary Register	Place the most significant 17 bits of the Q register in the least significant 17 bits of the store location specified by N. The most significant bit of the store location is made zero. The contents of A and Q are not affected by this function.
4	Read	Copy the contents of the store location specified by N into the accumulator. Q not affected.
5	Write	Copy the contents of the accumulator into the store location specified by N. A and Q not affected.
6	Collate *Nowadays usually called "Logical And".*	Place ones in the accumulator in only those digit positions in which both the contents of store location N and the contents of the accumulator are ones. Q not affected. (Alternatively: form in the accumulator the logical product of the contents of the accumulator and of the store location specified by N.)
7	Jump if zero	If the number in the accumulator is zero, place N in the sequence control register of the current program level. The contents of the accumulator are not affected by this function. Q is affected.
8	Jump	Place N in the sequence control register of the current program level. The contents of A and Q are not affected by this function.
9	Jump if negative	If the number in the accumulator is negative, place N in the sequence control register of the current program level. The contents of A are not affected by this function. Q is affected.
10	Count in Store	Increment the contents of the store location specified by N by $+2^{-17}$. The contents of A and Q are not affected by this function.
11	Store SCR	Copy the contents of the sequence control register of the current program level into the store location specified by N. The contents of A are not affected by this function. Q is affected*, in a defined (useful?) way*.

Function Number	Title	Specification
12	Multiply	Multiply the number in the accumulator by the number in the store location specified by N, and place the result in the accumulator and the most significant 17 digit positions of the Q register. The least significant digit of the Q register ~~is made zero~~ *after the multiplication contains the sign bit of the accumulator before the multiplication* ($Q_1 := A_{18}$).
13	Divide	Divide the number in the accumulator and the most significant 17 digits of the Q register by the number in the store location specified by N, and place the result in the accumulator, making the least significant digit a one. Q is affected.
14	Shift & block transfer	
a)	Left Shift $0 \le N \le 2047$ $b_{13}=0$ $b_{12}=0$	Shift the contents of the accumulator and Q register left by the number of places specified by N. (= multiply by 2^N).
b)	Right Shift $6144 \le N \le 8191$ $b_{13}=1$ $b_{12}=1$	Shift the contents of the accumulator and Q register right by 8192-N places, with sign bit regeneration. (= divide by 2^{8192-N}).
c)	Block Input $2048 \le N \le 4095$ $b_{13}=0$ $b_{12}=1$	Transfer n words of information from the device specified by bits 1 to 11 of N into store locations m to m+n-1, m is the contents of the accumulator and n is the contents of the Q register. ($N \le 4095$).
d)	Block Output $4096 \le N \le 6143$ $b_{13}=1$ $b_{12}=0$	Transfer n words of information to the device specified by bits 1 to 11 of N from store locations m to m+n-1 where m is the contents of the accumulator and n is the contents of the Q register.
15	Input/Output	
a)	Input $0 \le N \le 2047$	Input to the accumulator one 18 bit word from the device specified by bits 1 to 11 of N.
b)	Output $4096 \le N \le 6143$	Output from the accumulator one 18 bit word to the device specified by bits 1 to 11 of N.
c)	Tape reader input N=2048	Input one 7 *or 8* bit character from the tape reader channel. Shift the previous contents of the accumulator left by seven places and ~~*place*~~ *"or"* the input character in the least significant 7 *or 8* bit positions of the accumulator.
d)	Tape punch output N=6144	Output the least significant eight bits of the accumulator to the tape punch channel.
e)	Program Terminate N=7168	Terminate current program level.

This table omits the Teletype Input (15 2052) and Teletype Output (15 6148) instructions. These and other instructions relating to standard peripherals, including the paper-tape On-Line Adaptor, and the Second Paper Tape Station, deserve another section.

Instruction Modification.

The address specified by an instruction may be modified by adding the contents of the B register to the 13 address digits (N digits) of the instruction prior to implementing the instruction. Only the least significant 16 bits of the sum will be used as the address of the function. This does not affect the instruction held in the store. Where the instruction is required to be modified this is signified by a one in bit 18 position of the instruction. By this means it is possible to extend the range of store locations which may be specified by computer instructions to 65,536 locations and thereby obtain access to the additional store units.

Priority Level Program Organisation.

Each priority level has its own Sequence Control Register and B Register. *Nowadays, the SCR would be called "the Program Counter" (not that it is used to count programs), and the B Register would be called an "Index Register".* These registers are locations in store and can be referred to by programs in the normal way.

Priority Level		B. Reg. Location	S.C.R. Location
(Highest)	1	1	0
	2	3	2
	3	5	4
(Lowest)	4	7	6

The Accumulator and the Auxiliary Register are shared between all four levels, so they must be safeguarded by program every time an interrupt occurs. It will also usually be necessary to reset the Sequence Control Register on terminating a level so that the program, when next demanded, will start again at the same location.

All these conditions are fulfilled by the following control instructions. They are applicable to any program on levels 1, 2 or 3 which starts at location N.

		Instruction		
Location		**Function**	**Address**	**Remarks**
	(N−6)	—	—	Store for lower level AR.
	(N−5)	—	—	Store for lower level Acc.
	(N−4)	0	N−6	Reset lower level AR.
	(N−3)	14	1	*Shift left 1 place.*
	(N−2)	4	N−5	Reset lower level Acc.
	(N−1)	15	7168	Terminate, Note SCR reset to N.
Entry	(N)	5	N−5	Store lower level Acc.
	(N+1)	3	N−6	Store lower level AR.
	(N+)			Required program (x locations).
	(N+2+x)	8	N−4	Jump to reset for lower level.

If the contents of the AR on *any of* the lower level are not required then instructions N−4, N−3 and N+1 can be omitted and store location N−6 is not required.

The above control instructions are those quoted in reference [5]. Reference [4] contains the same explanatory text, and less efficient control instructions which reset the lower level AR by loading it into the accumulator and shifting right 17 places.

Note that this program does not preserve bit 1 of Q*, (the least significant bit of the AR).*

Any interrupts that occur during a Function 0 instruction are deferred. This make it safe (in this control code or elsewhere) to load the most significant 17 bits of the AR using function 0 and a one-place left shift, even though this briefly uses the least significant bit.

Programs entered by the JUMP button are obeyed from level 1. The initial program should set up locations 2, 4 and 6 to appropriate addresses, then terminate to level 4, which is regarded as the base level.

Initial Instructions.

Locations 8180 to 8191 inclusive are used for the initial instructions, used to input an initial program loader punched in "binary" form. These instructions are entered on level 1 at location 8181. On 903 processors fitted with more than 8192 words of core store the locations 8180 to 8191 may be used as normal core store when the initial instructions are "disabled". The instructions are disabled whenever a 15 7168 is obeyed. They are enabled whenever the JUMP button is pressed. The contents of 8180 to 8191 will be preserved unless program is obeyed from those locations. The effect of reading these locations on a basic machine or while the instructions are enabled should be regarded as undefined.

Address	**Instruction**			**Effect**
'N' digits	'B'	'F'	'N'	
8180	/	15	8189	(−3)
8181		0	8180	(Set B-Register to −3)
8182		4	8189	(Set Accumulator initially)
8183		15	2048	(Shift and input tape character)
8184		9	8186	(Jump to 8186 if Accumulator is negative)
8185		8	8183	(Jump to 8183 if Accumulator is positive)
8186		15	2048	(Shift and input final tape character of word)
8187	/	5	8180	(Store word read in)
8188		10	0001	(Count in B-Register)
8189		4	0001	(Read B-Register)
8190		9	8182	(Jump to 8182 if Accumulator is negative)
8191		8	8177	(Jump to 8177 if Accumulator is positive)

When entered at 8181 the routine initially reads words into 8177, 8178 and 8179, control is then transferred to location 8177. If these instructions set the B register to −n and then transfer control to 8182, words can then be read into the sequence of n locations ending at 8179. Control is then transferred again to location 8177 so that a transfer instruction read into that location can trigger the program*, or enter a "dynamic stop"*.

Store Addressing.

The basic 8192 words of store can be addressed directly by the 13 bit instruction address (N). Larger stores are considered to be divided into modules of 8192 words (Module numbers from 0 to 7). Within each module, program and data are addressed relative to the first location (location 0) of that module. B-modification is used to access program and data outside the current module. The address actually addressed is $(S_{16-14}+N+B)_{16-1}$ therefore the address used for modification must be relative to the current module. Attempts to address store modules not fitted cause the processor to hold up.

Because even the modified jumps used for subroutine exit are interpreted relative to the start of the module, Function 11, Store SCR, stores the module-relative address of the next instruction, namely just S_{13-1}, into the specified store location. The remaining bits of S are placed in Q, (to facilitate return from "far calls" to subroutines in other modules?).

920M.

Very similar to the 920B described in detail above, but with the following improvement, deduced from "Programming Compatibility of 920 Series Computers" [2]:

Function 9	Jump if Negative does not affect the Q-register.

The "Elliott MCS 920M Facts" card, [8], describes the 920M Order Code as having "Up to 60 functions", when in fact it has exactly the same set of "16 basic functions" and handful of sub-functions as the 920B. The 920M facts card has been "sexed up" (in current terminology), probably as a marketing ploy. For example, the basic functions, applied to the SCR and B-register in memory, are treated as distinct functions.

920C, 905, ARCH 9050.

Similar to the 920M described above, but with some significant improvements:

Function 7	Jump if Zero does not affect the Q-register.
Function 15	Nine extra instructions.
Store Addressing	Introduction of Absolute Addressing mode
Register Structure	The current level SCR is held in a Register, not in the Store.

The wording for sections "Order Code" and "Store Addressing" below is based on wording taken from the "Elliott MCS 920C Facts" card [12] and "Elliott 905 Computer Facts" card [13]. *Comments in italic are not part of those original texts.*

Both of these facts cards give the instruction set in "sexed up" form again. The 920C facts card takes this to extreme, listing 71 functions, by describing each function up to four times: as modified or not, and as absolute or relative. Function 0 occurs nine times, four times as "Load B", four times as "Load Q", and once as "B to Q".

Order Code.

The nine extra instructions follow on from the existing program terminate instruction:

Instruction	Title	Specification
15 7168	Terminate	Terminate current program level, and disable Initial Instructions.
15 7169	Test standard	Skip the next instruction if the contents of the accumulator are standardized (i.e. if A ≥ +½ or A < −½ or A = 0).
15 7170	Count and test	Increment the contents of the current level B register by $+2^{-17}$ and skip the next instruction if bits 1 to 13 of the result are zero.
15 7171	WG Input	Transfer the setting of the 18 word generator switches *(the "hand keys")* into the Accumulator.
15 7172	A to Aux	Place the least significant 17 bits of the Accumulator in the most significant 17 bits of the Q register. The least significant bit of the Q register is made zero.
15 7173	Aux to A	Place the most significant 17 bits of the Q register in the least significant 17 bits of the Accumulator. The most significant bit of the Accumulator is made zero.
15 7174	A to B	Place the contents of the Accumulator in the current level B register.
15 7175	B to A	Place the contents of the current level B register in the Accumulator.
15 7176	Set relative	Set Relative Addressing mode, H := 0.
15 7177	Set absolute	Set Absolute Addressing mode, H := 1, and disable Initial Instructions.

Store Addressing.

The store is divided into blocks of 8192 words each; numbers or instructions may be placed in any location of any available block. The N bits of an instruction are interpreted according to the function being obeyed and the state of the Address Mode Indicator (H).

If H = 0 or on functions 7, 8 and 9
: the N bits specify a location of the block in which the instruction itself is located; i.e. the absolute address of the location is formed by adding the N bits to bits 14-17 of the S register, the latter being those which specify the block in which the instruction is located.

If H = 1 but not on functions 7, 8 or 9
: the N bits specify absolutely a location in the first block of store (0 to 8191).

Thus, if H = 0, the 920C/905 behaves like a 920B/903, where all addresses are interpreted relative to the start of the current 8192-word store unit. If H = 1, jumps are still interpreted relative to the current store unit, but data addresses are absolute.

The address specified by an instruction may be modified by adding the contents of the B register to the un-modified address prior to implementing the instruction. This does not affect the instruction held in store.

Where the instruction is required to be modified this is signified by a one in bit 18 position of the instruction. Only the least significant 17 bits of the sum will be used as the address; by this means the range of storage locations which may be specified by computer instructions is increased to 131,072 words.

Thus, only 17 bits of the SCR are required. The 18th (most significant) bit of locations 0, 2, 4 and 6 is used to hold H (the Address Mode Indicator) for that level. *For backward compatibility, programs entered by the JUMP button commence (on level 1) in Relative Address mode.*

920ATC & MC1800.

The extra facilities offered by these later "spin-off" machines included:

B-Modification	B-Modification does not affect the Q-register.
Register Structure	The current level B-Register is held in a Register, not in the Store.
Floating Point	Introduction of a Floating Point Arithmetic mode

I'm not going any further with these in the Pilot Study.

Instruction Times.

Gathered together for ease of comparison, from various sources:

- The 920A times are taken from the facts card via [3].
- The 920B/903 times are taken from the 920B facts card [5], and almost agree with those in the newer 903 facts card [7]. The times quoted in the 920B brochure [4] and older 903 facts card [6] show more variation but are generally similar.
- The 5µsec 920M times are taken from the facts card [8] and brochure [9], which agree.
- The 2µsec 920M times are taken from the brochure [10].
- The 1µsec and 2µsec 905 times are taken from the facts card [13].
- The 1µsec 920C times given on the glued label in the facts card [12] vary slightly from the 1µsec 905 times shown below, whilst the times given behind the glued label agree with those given in the brochure [11] and are about 10% faster than those show.

There may be no such thing as a 2µsec 920C, (especially as a 920M has a 2µsec store).
Note how the times for the 2µsec 920M and 2µsec 905 differ substantially, due in part to the fact that the 920M keeps its Sequence Control Register in the store.

Instruction	Effect	Time in µsec					
		920A	920B or 903	920M	920M	*920C or* 905	920C or 905
	for a Store Cycle time of	*6-7½?*	6	5	2	2	1
/	For B Modification, add	8	6.5	6	3.2	2.2 *	1.2 *
0 N	Set B register	33	28.5	22	10.8	6.6	3.6
1 N	Add	27	24	19	10.6	4.4	2.4
2 N	Negate and Add	33	27	21	12.6	5.3	3.3
3 N	Store Auxiliary Register	27	23.5	20	11.6	5.3	3.3
4 N	Read	27	24	19	10.6	4.4	2.4
5 N	Write	27	23.5	20	11.6	5.3	3.3
6 N	Collate	30	23.5	19	10.6	4.4	2.4
7 N	Jump if zero A < 0	25	20	16	10.4	2.2 *	1.2 *
	Jump if zero A > 0	28	21.5	17	11.4	2.2 *	1.2 *
	Jump if zero A = 0	36	26.5	21	12.6	2.2	1.2
8 N	Jump	27	24	19	10.6	2.2	1.2
9 N	Jump if negative A ≥ 0	33	20	15	9.4	2.2 *	1.2 *
	Jump if negative A < 0	33	26	19	10.6	2.2	1.2
10 N	Count in Store	30	24	20	11.6	5.6	3.6
11 N	Store SCR	38	31.6	25	17.8	5.3	3.3
12 N	Multiply	186	76.5	38	29.6	12.2	10.2
13 N	Divide	192	79.5	39	30.6	21.3	19.3
14 0-36 and	Left Shift or Right Shift	28	22	16	10.4	3.9	2.9
14 8156-8191	n places	+9n	+3n	+n	+n	+0.9n	+0.9n
14 2048-6143	Block Input or Output	n/a	≥23.5	≥18	≥12.4	≥4.8	≥3.8
	n words		+9.5n	+10n	+7.2n	+6.5n	+5.5n
15 0-6144	Single Input or Output	≥28	≥20.5	≥20	≥14.4	≥6.5	≥5.5
n/a	Program Interrupt	0	0	0	0	*5.6?*	3.6
15 7168	Program Terminate	*28?*	20.5	20	14.4	11.4	7.4
15 7169	Test standard no skip	n/a	n/a	n/a	n/a	4.9	2.9
	Test standard skip	n/a	n/a	n/a	n/a	5.8	3.8
15 7170	Count and test no skip	n/a	n/a	n/a	n/a	8.3	5.3
	Count and test skip	n/a	n/a	n/a	n/a	9.2	6.2
15 7171	Key Input	n/a	n/a	n/a	n/a	5.7	3.7
15 7172	A to Aux	n/a	n/a	n/a	n/a	4.7	3.7
15 7173	Aux to A	n/a	n/a	n/a	n/a	4.7	3.7
15 7174	A to B	n/a	n/a	n/a	n/a	6.1	4.1
15 7175	B to A	n/a	n/a	n/a	n/a	6.1	4.1
15 7176	Set relative	n/a	n/a	n/a	n/a	5.7	3.7
15 7177	Set absolute	n/a	n/a	n/a	n/a	5.7	3.7

* On the 920C/905, modifying a conditional jump adds no time if the jump is not taken.

12/13-bit machines.

I'm not going to include these in the Pilot Study.

Terry Froggatt, February 2004.