### 2.6.1 Pipeline/Timing of Control Register Accesses

As shown in this section, all **MVC** are single-cycle instructions that complete their access of the explicitly named registers in the E1 pipeline phase. This is true whether **MVC** is moving a general register to a control register, or vice versa. In all cases the source register content is read, moved through the .S2 unit, and written to the destination register in the E1 pipeline phase.

| **Pipeline Stage** | **E1** |
|---|---|
| **Read** | *src2* |
| **Written** | *dst* |
| **Unit in use** | .S2 |

Even though **MVC** modifies the particular target control register in a single cycle, it can take extra clocks to complete modification of the non-explicitly named register. For example, the **MVC** cannot modify bits in the IFR directly. Instead, **MVC** can only write 1's into the ISR or the ICR to specify setting or clearing, respectively, of the IFR bits. **MVC** completes this ISR/ICR write in a single (E1) cycle (as described above) but the modification of the IFR bits themselves occur one clock later. For more information on the manipulation of ISR, ICR, and IFR see these control hardware sections in Chapter 8: section 8.3.2 *Status of, Setting, and Clearing Interrupts,* and section 8.3.3 *Returning from Interrupt Servicing Control Hardware..*

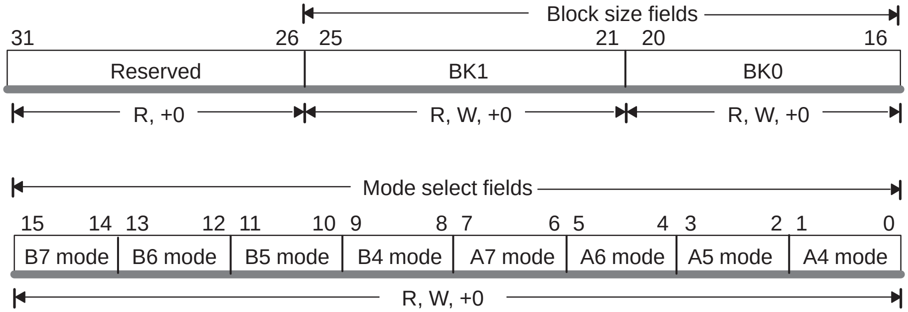
Saturating instructions, such as **SADD**, set the saturation flag bit (SAT) in the Control Status Register (CSR) indirectly. As a result, several of these instructions update the SAT bit one full clock cycle after their primary results are written to the register file. For example, the **SMPY** instruction writes its result at the end of pipeline stage E2; its primary result is available after one delay slot. In contrast, the SAT bit in the CSR is updated one cycle later than the result is written; this update occurs after two delay slots. (For the specific behavior of an instruction, refer to the description of that individual instruction).

The **B IRP** and **B NRP** instructions directly update the GIE and NMIE, respectively. Because these branches directly modify the CSR and IER (Interrupt Enable Register) respectively, there are no delay slots between when the branch is issued and when the control register updates take effect.

### 2.6.2 Addressing Mode Register (AMR)

For each of the eight registers (A4–A7, B4–B7) that can perform linear or circular addressing, the AMR specifies the addressing mode. A 2-bit field for each register selects the address modification mode: linear (the default) or circular mode. With circular addressing, the field also specifies which BK (block size) field to use for a circular buffer. In addition, the buffer must be aligned on a byte boundary equal to the block size. The mode select fields and block size fields are shown in Figure 2–5, and the mode select field encoding is shown in Table 2–4.

*Figure 2–5. Addressing Mode Register (AMR)*




**Legend**: R Readable by the **MVC** instruction
W Writeable by the MVC instruction
+0 Value is zero after reset

*Table 2–4. Addressing Mode Register (AMR) Mode Select Field Encoding*

| Mode | Description |
|---|---|
| 0 0 | Linear modification (default at reset) |
| 0 1 | Circular addressing using the BK0 field |
| 1 0 | Circular addressing using the BK1 field |
| 1 1 | Reserved |

The reserved portion of AMR is always 0. The AMR is initialized to 0 at reset.

The block size fields, BK0 and BK1, contain 5-bit values used in calculating block sizes for circular addressing.

*Block size (in bytes) =* $2^{(N+1)}$
where $N$ is the 5-bit value in BK0 or BK1

Table 2–5 shows block size calculations for all 32 possibilities.

*Table 2–5. Block Size Calculations*

| N | Block Size | N | Block Size |
|---|---|---|---|
| 00000 | 2 | 10000 | 131 072 |
| 00001 | 4 | 10001 | 262 144 |
| 00010 | 8 | 10010 | 524 288 |
| 00011 | 16 | 10011 | 1 048 576 |
| 00100 | 32 | 10100 | 2 097 152 |
| 00101 | 64 | 10101 | 4 194 304 |
| 00110 | 128 | 10110 | 8 388 608 |
| 00111 | 256 | 10111 | 16 777 216 |
| 01000 | 512 | 11000 | 33 554 432 |
| 01001 | 1 024 | 11001 | 67 108 864 |
| 01010 | 2 048 | 11010 | 134 217 728 |
| 01011 | 4 096 | 11011 | 268 435 456 |
| 01100 | 8 192 | 11100 | 536 870 912 |
| 01101 | 16 384 | 11101 | 1 073 741 824 |
| 01110 | 32 768 | 11110 | 2 147 483 648 |
| 01111 | 65 536 | 11111 | 4 294 967 296 |

**Note:** When N is 11111, the behavior is identical to linear addressing.