

Figure 6-4. The Parallel Instruction Fetches and Instruction Executions

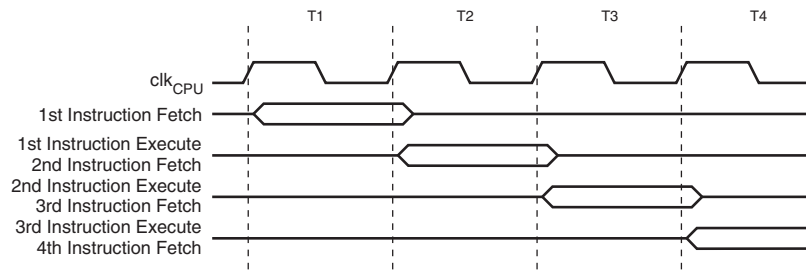
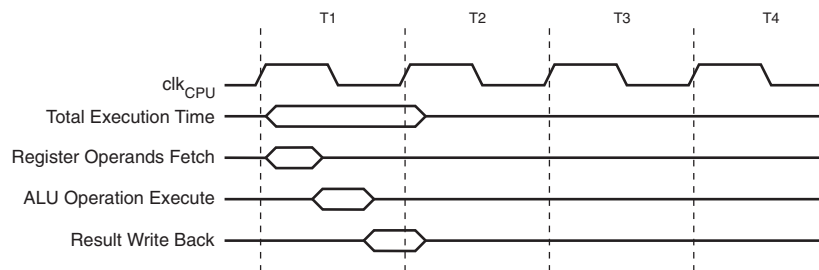


Figure 6-5 shows the internal timing concept for the Register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 6-5. Single Cycle ALU Operation



6.8 Reset and Interrupt Handling

The Atmel®AVR® provides several different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section “[Memory Programming](#)” on page 291 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt vectors. The complete list of vectors is shown in “[Interrupts](#)” on page 59. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The interrupt vectors can be moved to the start of the boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to “[Interrupts](#)” on page 59 for more information. The Reset vector can also be moved to the start of the boot Flash section by programming the BOTRST fuse, see “[Boot Loader Support – Read-While-Write Self-Programming](#)” on page 277.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the interrupt flag. For these interrupts, the Program Counter is vectored to the actual interrupt vector