Zero Day Exploit

Processor Specifications

# Instructions

|  |  |
| --- | --- |
| Instruction | Description |
| DSTR | Moves data from memory to the disk |
| DLDR | Moves data from the disk to memory |
| CTD | Counts the number of attached disks |
| CPM | Changes the processor mode |
| HLT | Halts the processor |
| PUSH | Pushes a value to the stack |
| POP | Pops a value from the stack |
| B | Performs a far jump, relative to the current execution position |
| BL | In addition to a far jump, also sets r14 to the previous execution position |
| BX | Performs an absolute far jump, while changing the processor mode |
| STR | Moves data from a register to memory, in integer form |
| LDR | Moves data from memory to a register, in integer form |
| STRB | Moves the least significant byte of a register to memory |
| LDRB | Moves a byte from memory into a register |
| SWP | Swaps an integer between memory into a register |
| SWPB | Swaps a byte between memory and a register |
| MUL | Multiplies two operands |
| SVC | Executes an interrupt, elevating the execution state in the process |
| EOR | Performs an exclusive or on two operands |
| SUB | Subtracts two operands |
| RSB | Reverse-subtracts two operands |
| ADD | Adds two operands |
| TST | Sets status flags based on the result of ANDing two operands |
| TEQ | Sets status flags based on the result of XORing two operands |
| CMP | Sets status flags based on the result of subtracting two operands |
| CMN | Sets status flags based on the result of adding two operands |
| ORR | Performs an or on two operands |
| MOV | Moves a value between registers |
| BIC | Clears the specified bits of a register |
| MVN | Moves a value between registers, performing a logical NOT on the second operand |
| AND | Performs an AND on two operands |

# Instruction Arguments

|  |  |
| --- | --- |
| Instruction | Argument Scheme |
| DSTR | Rmem\_start, Rdisk\_start, Rlength, ARGid |
| DLDR | Rmem\_start, Rdisk\_start, Rlength, ARGid |
| CTD | Rto\_store |
| CPM | IMMmode |
| HLT | N/A |
| PUSH | Rto\_push |
| POP | Rto\_pop |
| B | IMMto\_jump |
| BL | IMMto\_jump |
| BX | Rto\_jump, Rmode |
| STR | Rbase, Rto\_store, IMMoffset |
| LDR | Rbase, Rto\_store, IMMoffset |
| STRB | Rbase, Rto\_store, IMMoffset |
| LDRB | Rbase, Rto\_store, IMMoffset |
| SWP | Rbase, Rto\_store, IMMoffset |
| SWPB | Rbase, Rto\_store, IMMoffset |
| MUL | Rdest, Rop1, ARGop2 |
| SVC | IMMinterrupt |
| EOR | Rdest, Rop1, ARGop2 |
| SUB | Rdest, Rop1, ARGop2 |
| RSB | Rdest, Rop1, ARGop2 |
| ADD | Rdest, Rop1, ARGop2 |
| TST | Rop1, ARGop2 |
| TEQ | Rop1, ARGop2 |
| CMP | Rop1, ARGop2 |
| CMN | Rop1, ARGop2 |
| ORR | Rdest, Rop1, ARGop2 |
| MOV | Rdest, ARGop |
| BIC | Rdest, Rto\_clear |
| MVN | Rdest, ARGop |
| AND | Rdest, Rop1, ARGop2 |

# Mask / Data Pairs

|  |  |  |
| --- | --- | --- |
| Instruction | Unique Data | Bitmask |
| DSTR | 0000-1011-0000-0000-0000-0000-0000-0000 | 0000-1111-0000-0000-0000-0000-0000-0000 |
| DLDR | 0000-1001-0000-0000-0000-0000-0000-0000 | 0000-1111-0000-0000-0000-0000-0000-0000 |
| CTD | 0000-0101-1111-1111-1111-1111-1111-0000 | 0000-1111-1111-1111-1111-1111-1111-0000 |
| CPM | 0000-1111-1111-1111-1111-1111-1111-0000 | 0000-1111-1111-1111-1111-1111-1111-0000 |
| HLT | 0000-1111-1111-1111-1111-1111-1111-1111 | 0000-1111-1111-1111-1111-1111-1111-1111 |
| PUSH | 0000-1110-0111-1111-1111-1111-1111-0000 | 0000-1111-1111-1111-1111-1111-1111-0000 |
| POP | 0000-1111-1111-1111-1111-1111-1111-0000 | 0000-1111-1111-1111-1111-1111-1111-0000 |
| B | 0000-1010-0000-0000-0000-0000-0000-0000 | 0000-1111-0000-0000-0000-0000-0000-0000 |
| BL | 0000-1011-0000-0000-0000-0000-0000-0000 | 0000-1111-0000-0000-0000-0000-0000-0000 |
| BX | 0000-0001-0010-1111-1111-1111-0000-0000 | 0000-1111-1111-1111-1111-1111-0000-0000 |
| STR | 0000-0100-0000-0000-0000-0000-0000-0000 | 0000-1100-0101-0000-0000-0000-0000-0000 |
| LDR | 0000-0100-0001-0000-0000-0000-0000-0000 | 0000-1100-0101-0000-0000-0000-0000-0000 |
| STRB | 0000-0100-0100-0000-0000-0000-0000-0000 | 0000-1100-0101-0000-0000-0000-0000-0000 |
| LDRB | 0000-0100-0101-0000-0000-0000-0000-0000 | 0000-1100-0101-0000-0000-0000-0000-0000 |
| SWP | 0000-0001-0000-0000-0000-0000-1001-0000 | 0000-1111-1111-0000-0000-1111-1111-0000 |
| SWPB | 0000-0001-0100-0000-0000-0000-1001-0000 | 0000-1111-1111-0000-0000-1111-1111-0000 |
| MUL | 0000-0000-0000-0000-0000-0000-1001-0000 | 0000-0000-0010-0000-0000-0000-1111-0000 |
| SVC | 0000-1111-0000-0000-0000-0000-0000-0000 | 0000-1111-0000-0000-0000-0000-0000-0000 |
| EOR | 0000-0000-0010-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| SUB | 0000-0000-0100-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| RSB | 0000-0000-0110-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| ADD | 0000-0000-1000-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| TST | 0000-0001-0000-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| TEQ | 0000-0001-0010-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| CMP | 0000-0001-0100-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| CMN | 0000-0001-0110-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| ORR | 0000-0001-1000-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| MOV | 0000-0001-1010-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| BIC | 0000-0001-1100-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| MVN | 0000-0001-1110-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |
| AND | 0000-0000-0000-0000-0000-0000-0000-0000 | 0000-1101-1110-0000-0000-0000-0000-0000 |

# Condition Codes

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Binary Value | Flags | Description |
| EQ | 0000 | Z set | equal |
| NE | 0001 | Z clear | not equal |
| CS | 0010 | C set | unsigned higher or same |
| CC | 0011 | C clear | unsigned lower |
| MI | 0100 | N set | negative |
| PL | 0101 | N clear | positive or zero |
| VS | 0110 | V set | overflow |
| VC | 0111 | V clear | no overflow |
| HI | 1000 | C set and Z clear | unsigned higher |
| LS | 1001 | C clear or Z set | unsigned lower or same |
| GE | 1010 | N equals V | greater or equal |
| LT | 1011 | N not equal to V | less than |
| GT | 1100 | Z clear AND (N equals V) | greater than |
| LE | 1101 | Z set OR (N not equal to V) | less than or equal |
| AL | 1110 | (ignored) | always |

# Processor Modes

|  |  |  |
| --- | --- | --- |
| Name | Binary Value | Description |
| User (USR) | 1000 | Unprivileged mode, programs should run here |
| FIQ | 0100 | Fast interrupt |
| Abort (ABT) | 0010 | Entered after an exception |
| IRQ | 0001 | Normal interrupt |
| Undefined (UNDEF) | 1001 | Processor boots here |
| System (SYS) | 0011 | Privileged mode using user mode’s register set |