Instruction Set

David Farrell

March 2025

1 Micro-instructions

Special registers:

- `sr0` - program kounter `pk`,
- `sr1` - stack pointer `sp`,
- `sr2` - temp A `tmpA`,
- `sr3` - temp B `tmpB`,
- `sr4` - offset register `offs`.

Micro-instructions:

- `0000` - do nothing
- `0001` - begin instruction (and increment `pk`)
- `N002` - output `srN` to data bus
- `N102` - output `srN` to addr bus
- `N202` - output *`srN` to data bus
- `N302` - output *(`srN`+`offs`) to data bus
- `N402` - write to `srN` from data bus
- `2502` - output `tmpA` to `pk` via secret bus
- `3502` - output `tmpB` to `pk` via secret bus
- `0502` - increment `pk`
- `1502` - increment `sp`
- `1602` - decrement `sp`
- `0602` - write to `pk` from `tmpA` if data bus is zero; increment otherwise
- `0702` - write to `pk` from `tmpA` if data bus is nonzero; increment otherwise
- `0802` - write to `pk` from `tmpA` if data bus is negative; increment otherwise
- `0902` - write to `pk` from `tmpA` if data bus is non-negative; increment otherwise
- `0a02` - write to `pk` from `tmpA` if data bus is positive; increment otherwise
- `0b02` - write to `pk` from `tmpA` if data bus is non-positive; increment otherwise
- `N003` - output `rN` to data bus
- `N103` - output `rN` to addr bus
- `N203` - output *`rN` to data bus
- `N303` - output *(`rN`+`offs`) to data bus

- `N403` - write to `rN` from data bus
- `0004` - write data bus to *(addr bus)
- `0104` - write data bus to *(addr bus+`offs`)
- `0005` - output *(addr bus) RAM to data bus
- `0105` - output *(addr bus+`offs`) RAM to data bus
- `N006` - read N bytes from RAM and put on the data bus. N = 1,2,4.
- `NM1K` - output ALU operation K on (`rN`, `rM`) to data bus
- `0020` - echo data bus to itself for 2 cycles
- `0021` - echo data bus to addr bus for 3 cycles
- `0022` - echo addr bus to data bus for 2 cycles
- `0023` - echo addr bus to itself for 2 cycles
- `0024` - echo addr bus to itself for 3 cycles
- `0025` - echo data bus to itself for 3 cycles
- `0026` - echo addr bus to data bus for 3 cycles
- `fffe` - end instruction
- `ffff` - reset everything

2 Actual instructions

2.1 Control

- `0000` - `chill`. Do nothing. Go to next instruction. [implemented]
  - Micro-instruction sequence:
    - `fffe` - end instruction
- `0001` - `goto`. Write constant to program kounter. (GOTO)
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `0402` - write to `pk` from data bus
    - `fffe` - end instruction
- `N002` - `gotoz`. goto if register `N` is zero.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N003` - output `rN` to data bus
    - `0602` - write to `pk` from `tmpA` if data bus is zero; increment otherwise
    - `fffe` - end instruction
- `N003` - `gotonz`. goto if register `N` is nonzero.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N003` - output `rN` to data bus
    - `0702` - write to `pk` from `tmpA` if data bus is nonzero; increment otherwise
    - `fffe` - end instruction
- `N102` - `goton`. goto if register `N` is negative.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N003` - output `rN` to data bus
    - `0802` - write to `pk` from `tmpA` if data bus is negative; increment otherwise
    - `fffe` - end instruction
- `N103` - `gotonn`. goto if register `N` is non-negative.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N003` - output `rN` to data bus
    - `0902` - write to `pk` from `tmpA` if data bus is non-negative; increment otherwise
    - `fffe` - end instruction
- `N202` - `gotop`. goto if register `N` is positive.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N003` - output `rN` to data bus
    - `0a02` - write to `pk` from `tmpA` if data bus is positive; increment otherwise
    - `fffe` - end instruction

- `N203` - `gotonp`. goto if register `N` is non-positive.
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `2402` - write to `tmpA` from data bus
    * `N003` - output `rN` to data bus
    * `0b02` - write to `pk` from `tmpA` if data bus is non-positive; increment otherwise
    * `fffe` - end instruction

2.2 Arithmetic And Logic

- `NM1k` - Apply ALU operation k to registers `N` and `M` and save to register `M`.
  - Micro-instruction sequence:
    * `NM1k` - output ALU operation k on (`rN`, `rM`) to data bus
    * `N403` - write to `rN` from data bus
    * `fffe` - end instruction

2.3 Stack

- `0004` - `call`. push `pk` onto the stack and goto
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `2402` - write to `tmpA` from data bus
    * `0502` - increment `pk`
    * `1602` - decrement `sp`
    * `1102` - output `sp` to addr bus
    * `0002` - output `pk` to data bus
    * `0004` - write data bus to *(addr bus)
    * `2002` - output `tmpA` to data bus
    * `0402` - write to `pk` from data bus
    * `fffe` - end instruction
- `0005` - `return`. pop top of stack into `pk`.
  - Micro-instruction sequence:
    * `1202` - output *`sp` to data bus
    * `0402` - write to `pk` from data bus
    * `1502` - increment `sp`
    * `fffe` - end instruction
- `0006` - `push`. Just decrement `sp`
  - Micro-instruction sequence:
    * `1602` - decrement `sp`
    * `fffe` - end instruction
- `0106` - `push`. push constant onto the stack
  - Micro-instruction sequence:
    * `1602` - decrement `sp`
    * `0202` - output *`pk` to data bus
    * `0025` - echo data bus to itself for 3 cycles
    * `1102` - output `sp` to addr bus
    * `0004` - write data bus to *(addr bus)
    * `0502` - increment `pk`
    * `fffe` - end instruction

- `0206` - `push`. push *(constant) onto the stack
  - Micro-instruction sequence:
    - `1602` - decrement `sp`
    - `0202` - output *`pk` to data bus
    - `0021` - echo data bus to addr bus for 3 cycles
    - `0005` - output *(addr bus) RAM to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `1102` - output `sp` to addr bus
    - `0004` - write data bus to *(addr bus)
    - `0502` - increment `pk`
    - `fffe` - end instruction
- `N306` - `push`. push `rN` onto the stack
  - Micro-instruction sequence:
    - `1602` - decrement `sp`
    - `1102` - output `sp` to addr bus
    - `N003` - output `rN` to data bus
    - `0004` - write data bus to *(addr bus)
    - `fffe` - end instruction
- `N406` - `push`. push `*rN` onto the stack
  - Micro-instruction sequence:
    - `1602` - decrement `sp`
    - `N203` - output *`rN` to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `1102` - output `sp` to addr bus
    - `0004` - write data bus to *(addr bus)
    - `fffe` - end instruction
- `N506` - `push`. push `*rN+offs` onto the stack
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `4402` - write to `offs` from data bus
    - `1602` - decrement `sp`
    - `N303` - output *(`rN`+`offs`) to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `1102` - output `sp` to addr bus
    - `0104` - write data bus to *(addr bus+`offs`)
    - `fffe` - end instruction
- `N606` - `push`. push `srN` onto the stack
  - Micro-instruction sequence:
    - `1602` - decrement `sp`
    - `1102` - output `sp` to addr bus
    - `N002` - output `srN` to data bus
    - `0004` - write data bus to *(addr bus)
    - `fffe` - end instruction
- `N706` - `push`. push `*srN` onto the stack
  - Micro-instruction sequence:
    - `1602` - decrement `sp`
    - `N202` - output *`srN` to data bus

        * 0025 - echo data bus to itself for 3 cycles
        * 1102 - output sp to addr bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- N806 - push. push *srN+offs onto the stack
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 1602 - decrement sp
        * N302 - output *(srN+offs) to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * 1102 - output sp to addr bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- 0007 - pop. Just increment sp
    - Micro-instruction sequence:
        * 1502 - increment sp
        * fffe - end instruction

- 0207 - pop. pop top of stack into *(constant)
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 2402 - write to tmpA from data bus
        * 1202 - output *sp to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * 2102 - output tmpA to addr bus
        * 0004 - write data bus to *(addr bus)
        * 0502 - increment pk
        * 1502 - increment sp
        * fffe - end instruction

- N307 - pop. pop top of stack into rN.
    - Micro-instruction sequence:
        * 1202 - output *sp to data bus
        * N403 - write to rN from data bus
        * 1502 - increment sp
        * fffe - end instruction

- N407 - pop. pop top of stack into *rN.
    - Micro-instruction sequence:
        * 1202 - output *sp to data bus
        * N403 - write to rN from data bus
        * 1502 - increment sp
        * fffe - end instruction

2.4 Copying And Pasting

- `0030` - Set *(`constant`) to `constant`.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `0502` - increment `pk`
    - `0202` - output *`pk` to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `2102` - output `tmpA` to addr bus
    - `0004` - write data bus to *(addr bus)
    - `0502` - increment `pk`
    - `fffe` - end instruction
- `0130` - Set *(`constant`) to *(`constant`).
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `0502` - increment `pk`
    - `0202` - output *`pk` to data bus
    - `0021` - echo data bus to addr bus for 3 cycles
    - `0005` - output *(addr bus) RAM to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `2102` - output `tmpA` to addr bus
    - `0004` - write data bus to *(addr bus)
    - `0502` - increment `pk`
    - `fffe` - end instruction
- `N031` - Set *(`constant`) to `rN`.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `0021` - echo data bus to addr bus for 3 cycles
    - `N003` - output `rN` to data bus
    - `0004` - write data bus to *(addr bus)
    - `0502` - increment `pk`
    - `fffe` - end instruction
- `N131` - Set *(`constant`) to *`rN`.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N203` - output *`rN` to data bus
    - `0025` - echo data bus to itself for 3 cycles
    - `2102` - output `tmpA` to addr bus
    - `0004` - write data bus to *(addr bus)
    - `0502` - increment `pk`
    - `fffe` - end instruction
- `N231` - Set *(`constant`) to `*(rN+offs)`.
  - Micro-instruction sequence:
    - `0202` - output *`pk` to data bus
    - `2402` - write to `tmpA` from data bus
    - `N203` - output *`rN` to data bus

        * `0025` - echo data bus to itself for 3 cycles
        * `2102` - output `tmpA` to addr bus
        * `0004` - write data bus to *(addr bus)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N331` - Set *(`constant`) to `srN`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `0021` - echo data bus to addr bus for 3 cycles
        * `N003` - output `rN` to data bus
        * `0004` - write data bus to *(addr bus)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N431` - Set *(`constant`) to *`srN`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `2402` - write to `tmpA` from data bus
        * `N203` - output *`rN` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `2102` - output `tmpA` to addr bus
        * `0004` - write data bus to *(addr bus)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N531` - Set *(`constant`) to `*(srN+offs)`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `2402` - write to `tmpA` from data bus
        * `N203` - output *`rN` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `2102` - output `tmpA` to addr bus
        * `0004` - write data bus to *(addr bus)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N033` - Set `rN` to `constant`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `N403` - write to `rN` from data bus
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N034` - Set `rN` to *(`constant`).
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `2402` - write to `tmpA` from data bus
        * `2202` - output *`tmpA` to data bus
        * `N403` - write to `rN` from data bus
        * `0502` - increment `pk`
        * `fffe` - end instruction

- NM35 - Set rN to rM (i.e., copy from rM to rN).
  - Micro-instruction sequence:
    * N003 - output rN to data bus
    * M403 - write to rM from data bus
    * fffe - end instruction
- NM36 - Set rN to *rM.
  - Micro-instruction sequence:
    * M203 - output *rM to data bus
    * N403 - write to rN from data bus
    * fffe - end instruction
- NM37 - Set rN to *rM+offs.
  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 4402 - write to offs from data bus
    * 0502 - increment pk
    * M303 - output *(rM+offs) to data bus
    * N403 - write to rN from data bus
    * fffe - end instruction
- NM38 - Set rN to srM (i.e., copy from srM to rN).
  - Micro-instruction sequence:
    * N002 - output srN to data bus
    * M403 - write to rM from data bus
    * fffe - end instruction
- NM39 - Set rN to *srM.
  - Micro-instruction sequence:
    * M202 - output *srM to data bus
    * N403 - write to rN from data bus
    * fffe - end instruction
- NM3a - Set rN to *srM+offs.
  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 4402 - write to offs from data bus
    * M302 - output *(srM+offs) to data bus
    * N403 - write to rN from data bus
    * 0502 - increment pk
    * fffe - end instruction
- N03b - Set *rN to constant.
  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 2402 - write to tmpA from data bus
    * N103 - output rN to addr bus
    * 2002 - output tmpA to data bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction
- N03c - Set *rN to *(constant).

  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 2402 - write to tmpA from data bus
    * N103 - output rN to addr bus
    * 2202 - output *tmpA to data bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction

- NM3d - Set *rN to rM.
  - Micro-instruction sequence:
    * N103 - output rN to addr bus
    * M003 - output rM to data bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction

- NM3e - Set *rN to *rM.
  - Micro-instruction sequence:
    * M103 - output rM to addr bus
    * N003 - output rN to data bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction

- NM3f - Set *rN to *rM+offs.
  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 4402 - write to offs from data bus
    * M303 - output *(rM+offs) to data bus
    * 0025 - echo data bus to itself for 3 cycles
    * N103 - output rN to addr bus
    * 0004 - write data bus to *(addr bus)
    * 0502 - increment pk
    * fffe - end instruction

- NM40 - Set *rN to srM.
  - Micro-instruction sequence:
    * M002 - output srM to data bus
    * 0025 - echo data bus to itself for 3 cycles
    * N103 - output rN to addr bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction

- NM41 - Set *rN to *srM.
  - Micro-instruction sequence:
    * M202 - output *srM to data bus
    * 0025 - echo data bus to itself for 3 cycles
    * N103 - output rN to addr bus
    * 0004 - write data bus to *(addr bus)
    * fffe - end instruction

- NM42 - Set *rN to *srM+offs.
  - Micro-instruction sequence:
    * 0202 - output *pk to data bus
    * 4402 - write to offs from data bus

        * M303 - output *(rM+offs) to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N103 - output rN to addr bus
        * 0004 - write data bus to *(addr bus)
        * 0502 - increment pk
        * fffe - end instruction

- N043 - Set *(rN+offs) to constant.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 2402 - write to tmpA from data bus
        * N103 - output rN to addr bus
        * 2002 - output tmpA to data bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- N143 - Set *(rN+offs) to *(constant).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 2402 - write to tmpA from data bus
        * N103 - output rN to addr bus
        * 2202 - output *tmpA to data bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- NM44 - Set *(rN+offs) to rM.
    - Micro-instruction sequence:
        * N103 - output rN to addr bus
        * M003 - output rM to data bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- NM45 - Set *(rN+offs) to *rM.
    - Micro-instruction sequence:
        * M103 - output rM to addr bus
        * N003 - output rN to data bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- NM46 - Set *(rN+offs) to *(rM+offs2).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * M303 - output *(rM+offs) to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N103 - output rN to addr bus
        * 0004 - write data bus to *(addr bus)
        * 0502 - increment pk
        * fffe - end instruction

- NM47 - Set *(rN+offs) to srM.
    - Micro-instruction sequence:
        * M002 - output srM to data bus

        * 0025 - echo data bus to itself for 3 cycles
        * N103 - output rN to addr bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- NM48 - Set *(rN+offs) to *srM.
    - Micro-instruction sequence:
        * M202 - output *srM to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N103 - output rN to addr bus
        * 0004 - write data bus to *(addr bus)
        * fffe - end instruction

- NM49 - Set *(rN+offs1) to *(rM+offs2).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * M303 - output *(rM+offs) to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N103 - output rN to addr bus
        * 0004 - write data bus to *(addr bus)
        * 0502 - increment pk
        * fffe - end instruction

- N04a - Set srN to constant.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- N14a - Set srN to *constant.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM4b - Set srN to rM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk

        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM4c - Set srN to *rM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction
- NM4d - Set srN to *(rM+offs).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction
- NM4e - Set srN to srM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction
- NM4f - Set srN to *srM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk

        * `fffe` - end instruction

- `NM50` - Set `srN` to `*(srM+offs)`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `4402` - write to `offs` from data bus
        * `0502` - increment `pk`
        * `0202` - output *`pk` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `N102` - output `srN` to addr bus
        * `0104` - write data bus to *(addr bus+`offs`)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N051` - Set `*srN` to `constant`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `4402` - write to `offs` from data bus
        * `0502` - increment `pk`
        * `0202` - output *`pk` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `N102` - output `srN` to addr bus
        * `0104` - write data bus to *(addr bus+`offs`)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `N151` - Set `*srN` to `*constant`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `4402` - write to `offs` from data bus
        * `0502` - increment `pk`
        * `0202` - output *`pk` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `N102` - output `srN` to addr bus
        * `0104` - write data bus to *(addr bus+`offs`)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `NM52` - Set `*srN` to `rM`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus
        * `4402` - write to `offs` from data bus
        * `0502` - increment `pk`
        * `0202` - output *`pk` to data bus
        * `0025` - echo data bus to itself for 3 cycles
        * `N102` - output `srN` to addr bus
        * `0104` - write data bus to *(addr bus+`offs`)
        * `0502` - increment `pk`
        * `fffe` - end instruction

- `NM53` - Set `*srN` to `*rM`.
    - Micro-instruction sequence:
        * `0202` - output *`pk` to data bus

        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM54 - Set *srN to *(rM+offs).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM55 - Set *srN to srM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM56 - Set *srN to *srM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM57 - Set *srN to *(srM+offs).
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus

        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- N058 - Set *(srN+offs) to constant.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- N059 - Set *(srN+offs) to *constant.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM5a - Set *(srN+offs) to rM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM5b - Set *(srN+offs) to *rM.
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 4402 - write to offs from data bus
        * 0502 - increment pk
        * 0202 - output *pk to data bus
        * 0025 - echo data bus to itself for 3 cycles
        * N102 - output srN to addr bus
        * 0104 - write data bus to *(addr bus+offs)
        * 0502 - increment pk
        * fffe - end instruction

- NM5c - Set *(srN+offs1) to *(rM+offs2).

  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `4402` - write to `offs` from data bus
    * `0502` - increment `pk`
    * `0202` - output *`pk` to data bus
    * `0025` - echo data bus to itself for 3 cycles
    * `N102` - output `srN` to addr bus
    * `0104` - write data bus to *(addr bus+`offs`)
    * `0502` - increment `pk`
    * `fffe` - end instruction

- `NM5d` - Set `*(srN+offs)` to `srM`.
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `4402` - write to `offs` from data bus
    * `0502` - increment `pk`
    * `0202` - output *`pk` to data bus
    * `0025` - echo data bus to itself for 3 cycles
    * `N102` - output `srN` to addr bus
    * `0104` - write data bus to *(addr bus+`offs`)
    * `0502` - increment `pk`
    * `fffe` - end instruction

- `NM5e` - Set `*(srN+offs)` to `*srM`.
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `4402` - write to `offs` from data bus
    * `0502` - increment `pk`
    * `0202` - output *`pk` to data bus
    * `0025` - echo data bus to itself for 3 cycles
    * `N102` - output `srN` to addr bus
    * `0104` - write data bus to *(addr bus+`offs`)
    * `0502` - increment `pk`
    * `fffe` - end instruction

- `NM5f` - Set `*(srN+offs1)` to `*(srM+offs2)`.
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus
    * `4402` - write to `offs` from data bus
    * `0502` - increment `pk`
    * `0202` - output *`pk` to data bus
    * `0025` - echo data bus to itself for 3 cycles
    * `N102` - output `srN` to addr bus
    * `0104` - write data bus to *(addr bus+`offs`)
    * `0502` - increment `pk`
    * `fffe` - end instruction

3 I/O

- `0xN0a0` - Read 1bit port `constant` into `rN` (normal addressing)
  - Micro-instruction sequence:
    * `0202` - output *`pk` to data bus

        * 0021 - echo data bus to addr bus for 3 cycles
        * 00a0 - read 1bit port addr (normal addressing) to the data line
        * N403 - write to rN from data bus 0502 - increment pk
        * fffe - end instruction

- 0xN0a1 - Write rN to 1bit port constant (normal addressing)
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 0021 - echo data bus to addr bus for 3 cycles
        * N003 - output rN to data bus
        * 00a1 - write data line to 1bit port addr (normal addressing)0502 - increment pk
        * fffe - end instruction

- 0xN0a2 - Read 1bit ports constant into rN (bitwise addressing)
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 0021 - echo data bus to addr bus for 3 cycles
        * 00a2 - read 1bit port addr (bitwise addressing) to the data line
        * N403 - write to rN from data bus 0502 - increment pk
        * fffe - end instruction

- 0xN0a3 - Write rN to 1bit ports constant (bitwise addressing)
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 0021 - echo data bus to addr bus for 3 cycles
        * N003 - output rN to data bus
        * 00a3 - write data line to 1bit port addr (bitwise addressing)0502 - increment pk
        * fffe - end instruction

- 0xN0a4 - Read 16bit port constant into rN
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 0021 - echo data bus to addr bus for 3 cycles
        * 00a4 - read 16bit port addr to the data line
        * N403 - write to rN from data bus 0502 - increment pk
        * fffe - end instruction

- 0xN0a5 - Write rN to 16bit port constant
    - Micro-instruction sequence:
        * 0202 - output *pk to data bus
        * 0021 - echo data bus to addr bus for 3 cycles
        * N003 - output rN to data bus
        * 00a4 - write data line to 16bit port addr0502 - increment pk
        * fffe - end instruction

- 0xNMa6 - Read 1bit port rM into rN (normal addressing)
    - Micro-instruction sequence:
        * M103 - output rM to addr bus
        * 00a0 - read 1bit port addr (normal addressing) to the data line
        * N403 - write to rN from data bus
        * fffe - end instruction

- 0xNMa7 - Write rN to 1bit port rM (normal addressing)

  - Micro-instruction sequence:
    * `M103` - output `rM` to addr bus
    * `N003` - output `rN` to data bus
    * `00a1` - write data line to 1bit port `addr` (normal addressing)
    * `fffe` - end instruction

- `0xNMa8` - Read 1bit ports `rM` into `rN` (bitwise addressing)
  - Micro-instruction sequence:
    * `M103` - output `rM` to addr bus
    * `00a2` - read 1bit port `addr` (bitwise addressing) to the data line
    * `N403` - write to `rN` from data bus
    * `fffe` - end instruction

- `0xN1a9` - Write `rN` to 1bit ports `rM` (bitwise addressing)
  - Micro-instruction sequence:
    * `M103` - output `rM` to addr bus
    * `N003` - output `rN` to data bus
    * `00a3` - write data line to 1bit port `addr` (bitwise addressing)
    * `fffe` - end instruction

- `0xN0aa` - Read 16bit port `rM` into `rN`
  - Micro-instruction sequence:
    * `M103` - output `rM` to addr bus
    * `00a4` - read 16bit port `addr` to the data line
    * `N403` - write to `rN` from data bus
    * `fffe` - end instruction

- `0xN0ab` - Write `rN` to 16bit port `tM`
  - Micro-instruction sequence:
    * `M103` - output `rM` to addr bus
    * `N003` - output `rN` to data bus
    * `00a4` - write data line to 16bit port `addr0502` - increment `pk`
    * `fffe` - end instruction