

Instruction Set

David Farrell

March 2025

Special registers:

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

1 instructions

1.1 Control

- 0000 - **chill**. Do nothing. Go to next instruction. [implemented]
 - Micro-instruction sequence:
 - * 0xfffe - end instruction
- 0001 - **goto**. Write constant to program kounter. (GOTO)
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0006 - write to **pk** from data bus
 - * 0xfffe - end instruction
- N002 - **gotoz**. goto if register N is zero.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0014 - write to **pk** from **tmpA** if data bus is zero; increment otherwise
 - * 0xfffe - end instruction
- N003 - **gotonz**. goto if register N is nonzero.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0015 - write to **pk** from **tmpA** if data bus is nonzero; increment otherwise
 - * 0xfffe - end instruction
- N102 - **goton**. goto if register N is negative.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus

- * 0xN020 - output **rN** to data bus
 - * 0x0016 - write to **pk** from **tmpA** if data bus is negative; increment otherwise
 - * 0xfffe - end instruction
- N103 - **gotonn**. goto if register **N** is non-negative.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0017 - write to **pk** from **tmpA** if data bus is non-negative; increment otherwise
 - * 0xfffe - end instruction
- N202 - **gotop**. goto if register **N** is positive.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0018 - write to **pk** from **tmpA** if data bus is positive; increment otherwise
 - * 0xfffe - end instruction
- N203 - **gotonp**. goto if register **N** is non-positive.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0019 - write to **pk** from **tmpA** if data bus is non-positive; increment otherwise
 - * 0xfffe - end instruction

1.2 Arithmetic And Logic

- NM1k - Apply ALU operation **k** to registers **N** and **M** and save to register **M**.
 - Micro-instruction sequence:
 - * 0xNM4 - output ALU operation on (**rN**, **rM**) to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0xfffe - end instruction

1.3 Stack

- 0004 - **call**. push **pk** onto the stack and goto
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0x0010 - increment **pk**
 - * 0x1011 - decrement **sp**
 - * 0x1003 - output **sp** to addr bus
 - * 0x0002 - output **pk** to data bus
 - * 0x0025 - write data bus to ***(addr bus)**
 - * 0x2002 - output **tmpa** to data bus
 - * 0x0006 - write to **pk** from data bus
 - * 0xfffe - end instruction
- 0005 - **return**. pop top of stack into **pk**.
 - Micro-instruction sequence:

- * 0x1005 - output $*(sp+offs)$ to data bus
 - * 0x0006 - write to **pk** from data bus
 - * 0x1010 - increment **sp**
 - * 0xfffe - end instruction
- 0006 - push. Just decrement **sp**
 - Micro-instruction sequence:
 - * 0x1011 - decrement **sp**
 - * 0xfffe - end instruction
- 0106 - push. push constant onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement **sp**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output **sp** to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment **pk**
 - * 0xfffe - end instruction
- 0206 - push. push $*(constant)$ onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement **sp**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0053 - move the value on the data bus to the addr bus
 - * 0x0027 - output $*(addr\ bus)$ RAM to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output **sp** to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment **pk**
 - * 0xfffe - end instruction
- N306 - push. push **rN** onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement **sp**
 - * 0x1003 - output **sp** to addr bus
 - * 0xN020 - output **rN** to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- N406 - push. push ***rN** onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement **sp**
 - * 0xN022 - output ***rN** to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output **sp** to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- N506 - push. push ***rN+offs** onto the stack
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus

- * 0x1011 - decrement `sp`
 - * 0xN023 - output `*(rN+offs)` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output `sp` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0xfffe - end instruction
- N606 - push. push `srN` onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement `sp`
 - * 0x1003 - output `sp` to addr bus
 - * 0xN002 - output `srN` to data bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0xfffe - end instruction
- N706 - push. push `*srN` onto the stack
 - Micro-instruction sequence:
 - * 0x1011 - decrement `sp`
 - * 0xN004 - output `*srN` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output `sp` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0xfffe - end instruction
- N806 - push. push `*srN+offs` onto the stack
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x1011 - decrement `sp`
 - * 0xN005 - output `*(srN+offs)` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x1003 - output `sp` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0xfffe - end instruction
- 0007 - pop. Just increment `sp`
 - Micro-instruction sequence:
 - * 0x1010 - increment `sp`
 - * 0xfffe - end instruction
- 0207 - pop. pop top of stack into `*(constant)`
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x2006 - write to `tmpa` from data bus
 - * 0x1005 - output `*(sp+offs)` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output `tmpa` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0x0010 - increment `pk`
 - * 0x1010 - increment `sp`
 - * 0xfffe - end instruction
- N307 - pop. pop top of stack into `rN`.

- Micro-instruction sequence:
 - * 0x1005 - output $*(sp+offs)$ to data bus
 - * 0xN024 - write to rN from data bus
 - * 0x1010 - increment sp
 - * 0xffffe - end instruction
- N407 - pop. pop top of stack into $*rN$.
 - Micro-instruction sequence:
 - * 0x1005 - output $*(sp+offs)$ to data bus
 - * 0xN024 - write to rN from data bus
 - * 0x1010 - increment sp
 - * 0xffffe - end instruction

1.4 Copying And Pasting

- 0030 - Set $*(constant)$ to $constant$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x2006 - write to $tmpa$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output $tmpa$ to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment pk
 - * 0xffffe - end instruction
- 0130 - Set $*(constant)$ to $*(constant)$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x2006 - write to $tmpa$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0053 - move the value on the data bus to the addr bus
 - * 0x0027 - output $*(addr\ bus)$ RAM to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output $tmpa$ to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment pk
 - * 0xffffe - end instruction
- N031 - Set $*(constant)$ to rN .
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0053 - move the value on the data bus to the addr bus
 - * 0xN020 - output rN to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment pk
 - * 0xffffe - end instruction
- N131 - Set $*(constant)$ to $*rN$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk

- * 0x2006 - write to `tmpa` from data bus
- * 0xN022 - output `*rN` to data bus
- * 0x0050 - prevent data bus from updating on next rising edge
- * 0x2003 - output `tmpa` to addr bus
- * 0x0025 - write data bus to `*(addr bus)`
- * 0x0010 - increment `pk`
- * 0xfffe - end instruction
- N231 - Set `*(constant)` to `*(rN+offs)`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x2006 - write to `tmpa` from data bus
 - * 0xN022 - output `*rN` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output `tmpa` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N331 - Set `*(constant)` to `srN`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0053 - move the value on the data bus to the addr bus
 - * 0xN020 - output `rN` to data bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N431 - Set `*(constant)` to `*srN`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x2006 - write to `tmpa` from data bus
 - * 0xN022 - output `*rN` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output `tmpa` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N531 - Set `*(constant)` to `*(srN+offs)`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x2006 - write to `tmpa` from data bus
 - * 0xN022 - output `*rN` to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0x2003 - output `tmpa` to addr bus
 - * 0x0025 - write data bus to `*(addr bus)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N033 - Set `rN` to `constant`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`

- * 0xN024 - write to **rN** from data bus
 - * 0x0010 - increment **pk**
 - * 0xffffe - end instruction
- N034 - Set **rN** to ***(constant)**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x2006 - write to **tmpa** from data bus
 - * 0x2005 - output ***(tmpa+offs)** to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0x0010 - increment **pk**
 - * 0xffffe - end instruction
- NM35 - Set **rN** to **rM** (i.e., copy from **rM** to **rN**).
 - Micro-instruction sequence:
 - * 0xN020 - output **rN** to data bus
 - * 0xM024 - write to **rM** from data bus
 - * 0xffffe - end instruction
- NM36 - Set **rN** to ***rM**.
 - Micro-instruction sequence:
 - * 0xM022 - output ***rM** to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0xffffe - end instruction
- NM37 - Set **rN** to ***rM+offs**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0xM023 - output ***(rM+offs)** to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0xffffe - end instruction
- NM38 - Set **rN** to **srM** (i.e., copy from **srM** to **rN**).
 - Micro-instruction sequence:
 - * 0xN002 - output **srN** to data bus
 - * 0xM024 - write to **rM** from data bus
 - * 0xffffe - end instruction
- NM39 - Set **rN** to ***srM**.
 - Micro-instruction sequence:
 - * 0xM004 - output ***srM** to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0xffffe - end instruction
- NM3a - Set **rN** to ***srM+offs**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0xM005 - output ***(srM+offs)** to data bus
 - * 0xN024 - write to **rN** from data bus
 - * 0x0010 - increment **pk**

- * 0xfffe - end instruction
- N03b - Set *rN to constant.
 - Micro-instruction sequence:
 - * 0x0011 - output *pk to data bus and increment pk
 - * 0x2006 - write to tmpa from data bus
 - * 0xN021 - output rN to addr bus
 - * 0x2002 - output tmpa to data bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0xfffe - end instruction
- N03c - Set *rN to *(constant).
 - Micro-instruction sequence:
 - * 0x0011 - output *pk to data bus and increment pk
 - * 0x2006 - write to tmpa from data bus
 - * 0xN021 - output rN to addr bus
 - * 0x2005 - output *(tmpa+offs) to data bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0xfffe - end instruction
- NM3d - Set *rN to rM.
 - Micro-instruction sequence:
 - * 0xN021 - output rN to addr bus
 - * 0xM020 - output rM to data bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0xfffe - end instruction
- NM3e - Set *rN to *rM.
 - Micro-instruction sequence:
 - * 0xM021 - output rM to addr bus
 - * 0xN020 - output rN to data bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0xfffe - end instruction
- NM3f - Set *rN to *rM+offs.
 - Micro-instruction sequence:
 - * 0x0011 - output *pk to data bus and increment pk
 - * 0x4006 - write to offs from data bus
 - * 0xM023 - output *(rM+offs) to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output rN to addr bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- NM40 - Set *rN to srM.
 - Micro-instruction sequence:
 - * 0xM002 - output srM to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output rN to addr bus
 - * 0x0025 - write data bus to *(addr bus)
 - * 0xfffe - end instruction

- NM41 - Set $*rN$ to $*srM$.
 - Micro-instruction sequence:
 - * 0xM004 - output $*srM$ to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output rN to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- NM42 - Set $*rN$ to $*srM+offs$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus
 - * 0xM023 - output $*(rM+offs)$ to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output rN to addr bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- N043 - Set $*(rN+offs)$ to constant.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x2006 - write to $tmpa$ from data bus
 - * 0xN021 - output rN to addr bus
 - * 0x2002 - output $tmpa$ to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- N143 - Set $*(rN+offs)$ to $*(constant)$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x2006 - write to $tmpa$ from data bus
 - * 0xN021 - output rN to addr bus
 - * 0x2005 - output $*(tmpa+offs)$ to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- NM44 - Set $*(rN+offs)$ to rM .
 - Micro-instruction sequence:
 - * 0xN021 - output rN to addr bus
 - * 0xM020 - output rM to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- NM45 - Set $*(rN+offs)$ to $*rM$.
 - Micro-instruction sequence:
 - * 0xM021 - output rM to addr bus
 - * 0xN020 - output rN to data bus
 - * 0x0025 - write data bus to $*(addr\ bus)$
 - * 0xfffe - end instruction
- NM46 - Set $*(rN+offs)$ to $*(rM+offs2)$.

- Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0xM023 - output ***(rM+offs)** to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output **rN** to addr bus
 - * 0x0025 - write data bus to ***(addr bus)**
 - * 0x0010 - increment **pk**
 - * 0xfffe - end instruction
- NM47 - Set ***(rN+offs)** to **srM**.
 - Micro-instruction sequence:
 - * 0xM002 - output **srM** to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output **rN** to addr bus
 - * 0x0025 - write data bus to ***(addr bus)**
 - * 0xfffe - end instruction
- NM48 - Set ***(rN+offs)** to ***srM**.
 - Micro-instruction sequence:
 - * 0xM004 - output ***srM** to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output **rN** to addr bus
 - * 0x0025 - write data bus to ***(addr bus)**
 - * 0xfffe - end instruction
- NM49 - Set ***(rN+offs1)** to ***(rM+offs2)**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0xM023 - output ***(rM+offs)** to data bus
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN021 - output **rN** to addr bus
 - * 0x0025 - write data bus to ***(addr bus)**
 - * 0x0010 - increment **pk**
 - * 0xfffe - end instruction
- N04a - Set **srN** to **constant**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output **srN** to addr bus
 - * 0x0026 - write data bus to ***(addr bus+offs)**
 - * 0x0010 - increment **pk**
 - * 0xfffe - end instruction
- N14a - Set **srN** to ***constant**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus

- * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- NM4b - Set `srN` to `rM`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
 - NM4c - Set `srN` to `*rM`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
 - NM4d - Set `srN` to `*(rM+offs)`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
 - NM4e - Set `srN` to `srM`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`

- * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- NM4f - Set `srN` to `*srM`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- NM50 - Set `srN` to `*(srM+offs)`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- N051 - Set `*srN` to constant.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- N151 - Set `*srN` to `*constant`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xffffe - end instruction
- NM52 - Set `*srN` to `rM`.
 - Micro-instruction sequence:

- * 0x0011 - output ***pk** to data bus and increment **pk**
- * 0x4006 - write to **offs** from data bus
- * 0x0010 - increment **pk**
- * 0x0011 - output ***pk** to data bus and increment **pk**
- * 0x0050 - prevent data bus from updating on next rising edge
- * 0xN003 - output **srN** to addr bus
- * 0x0026 - write data bus to ***(addr bus+offs)**
- * 0x0010 - increment **pk**
- * 0xffffe - end instruction
- NM53 - Set ***srN** to ***rM**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output **srN** to addr bus
 - * 0x0026 - write data bus to ***(addr bus+offs)**
 - * 0x0010 - increment **pk**
 - * 0xffffe - end instruction
- NM54 - Set ***srN** to ***(rM+offs)**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output **srN** to addr bus
 - * 0x0026 - write data bus to ***(addr bus+offs)**
 - * 0x0010 - increment **pk**
 - * 0xffffe - end instruction
- NM55 - Set ***srN** to **srM**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output **srN** to addr bus
 - * 0x0026 - write data bus to ***(addr bus+offs)**
 - * 0x0010 - increment **pk**
 - * 0xffffe - end instruction
- NM56 - Set ***srN** to ***srM**.
 - Micro-instruction sequence:
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x4006 - write to **offs** from data bus
 - * 0x0010 - increment **pk**
 - * 0x0011 - output ***pk** to data bus and increment **pk**
 - * 0x0050 - prevent data bus from updating on next rising edge

- * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- NM57 - Set `*srN` to `*(srM+offs)`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N058 - Set `*(srN+offs)` to constant.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- N059 - Set `*(srN+offs)` to `*constant`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction
- NM5a - Set `*(srN+offs)` to `rM`.
 - Micro-instruction sequence:
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x4006 - write to `offs` from data bus
 - * 0x0010 - increment `pk`
 - * 0x0011 - output `*pk` to data bus and increment `pk`
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output `srN` to addr bus
 - * 0x0026 - write data bus to `*(addr bus+offs)`
 - * 0x0010 - increment `pk`
 - * 0xfffe - end instruction

- NM5b - Set $*(srN+offs)$ to $*rM$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output srN to addr bus
 - * 0x0026 - write data bus to $*(addr\ bus+offs)$
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- NM5c - Set $*(srN+offs1)$ to $*(rM+offs2)$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output srN to addr bus
 - * 0x0026 - write data bus to $*(addr\ bus+offs)$
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- NM5d - Set $*(srN+offs)$ to srM .
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output srN to addr bus
 - * 0x0026 - write data bus to $*(addr\ bus+offs)$
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- NM5e - Set $*(srN+offs)$ to $*srM$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus
 - * 0x0010 - increment pk
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x0050 - prevent data bus from updating on next rising edge
 - * 0xN003 - output srN to addr bus
 - * 0x0026 - write data bus to $*(addr\ bus+offs)$
 - * 0x0010 - increment pk
 - * 0xfffe - end instruction
- NM5f - Set $*(srN+offs1)$ to $*(srM+offs2)$.
 - Micro-instruction sequence:
 - * 0x0011 - output $*pk$ to data bus and increment pk
 - * 0x4006 - write to $offs$ from data bus

- * 0x0010 - increment **pk**
- * 0x0011 - output ***pk** to data bus and increment **pk**
- * 0x0050 - prevent data bus from updating on next rising edge
- * 0xN003 - output **srN** to addr bus
- * 0x0026 - write data bus to ***(addr bus+offs)**
- * 0x0010 - increment **pk**
- * 0xfffe - end instruction

2 I/O

- 00a0 - Set I/O pin N to input mode.
 - Micro-instruction sequence:
 - * 0xN030 - set I/O pin N to input mode
- 00a1 - Set I/O pin N to output mode.
 - Micro-instruction sequence:
 - * 0xN031 - set I/O pin N to output mode
- 00a2 - Set I/O pin N to low.
 - Micro-instruction sequence:
 - * 0xN032 - set I/O pin N to low
- 00a3 - Set I/O pin N to high.
 - Micro-instruction sequence:
 - * 0xN033 - set I/O pin N to high
- NMa4 - Read I/O pin N to register M.
 - Micro-instruction sequence:
 - * 0xN034 - output I/O pin N to data bus
 - * 0xM024 - write to **rM** from data bus
- NMa5 - Write register M to I/O pin N.
 - Micro-instruction sequence:
 - * 0xM020 - output **rM** to data bus
 - * 0xN035 - write data bus to I/O pin N