

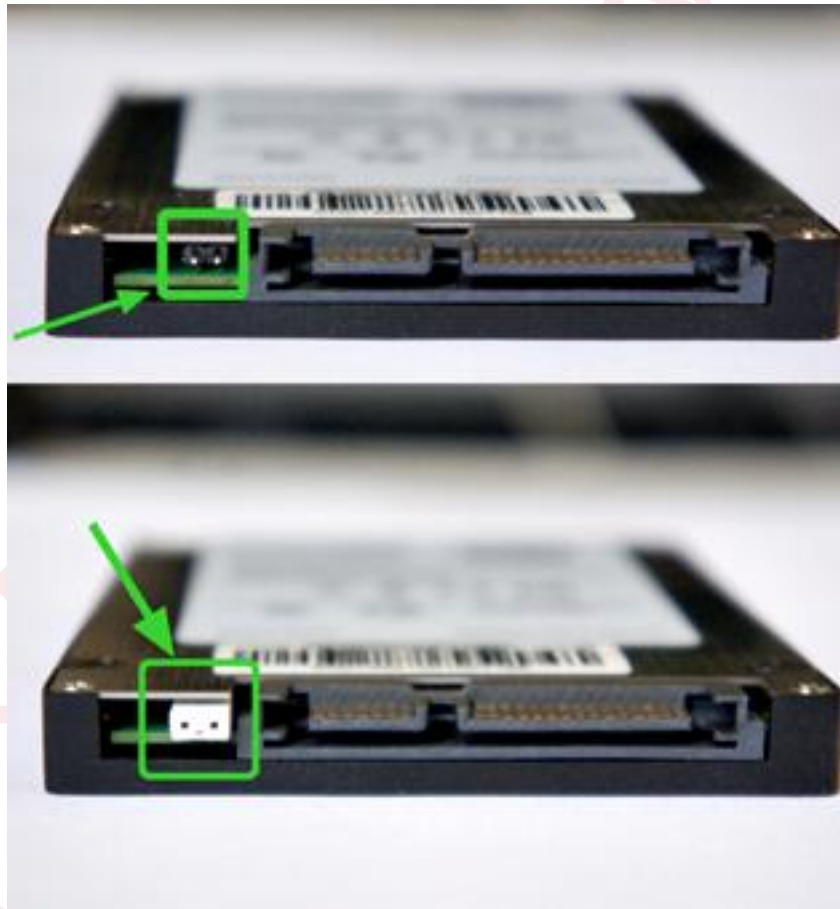
Recommendations on using Indilinx.Barefoot utility:

Indilinx.Barefoot utility features for working with SSD based on Indilinx.Barefoot controller in technological mode.

Controller name: **IDX100M0[X]**, where [X] = {0, 1, 2}

1. Technological mode.

The utility is available for working only if SSD is transferred in technological mode. For it, user should close contacts on PCB with jumper. The picture below shows a typical* arrangement of contacts.



*Some drives have another arrangement of PCB contacts.

Also, SSD can automatically go into technological mode during some critical fault. This mode is characterized by the following behavior:

a) Drive is detected as:

Model: YATAPDONG BAREFOOT-ROM

Serial: 0123456789ab (next part is not important)

Firmware: 00.P80

b) The drive returns the volume of 128 GB, regardless of the real capacity.

2. Utility

After launching the utility, drive will be initialized and searched for translator tables. It will take a few minutes. After initialization, information about current drive will be posted in log.

a) Bank configuration

SSD drives based on Indilinx Barefoot controller use four 16-bit channels. The diagram below shows the distribution of banks by channels.

Ch A ---Bank 0 --- Bank 4 --- Bank 8 ---- Bank 12 --- Bank 16 --- Bank 20 --- Bank 24 --- Bank 28

Ch B ---Bank 1 --- Bank 5 --- Bank 9 ---- Bank 13 --- Bank 17 --- Bank 21 --- Bank 25 --- Bank 29

Ch C ---Bank 2 --- Bank 6 --- Bank 10 --- Bank 14 --- Bank 18 --- Bank 22 --- Bank 26 --- Bank 30

Ch A ---Bank 3 --- Bank 7 --- Bank 11 --- Bank 15 --- Bank 19 --- Bank 23 --- Bank 27 --- Bank 31

Each bank consists of a pair of chips. One of the memory chips in the Bank is conventionally called the Low-chip, the second one is High-chip. The controller incorporates a couple of chips in one virtual chip with a 16-bit interface. During reading or writing operations all data alternate through the byte. For example: if we write some part of data into bank (0x0A, 0x2B, 0x0C, 0x11, 0x0E, 0x33...), data will be distributed among chips by the following way:

Low-chip: 0x0A, 0x0C, 0x0E ...

High-chip: 0x2B, 0x11, 0x33 ...

After utility launching, information about banks (which banks are involved and its whole count) will be posted into log. For each involved banks information about Low-chip and High-chip appears in the log.

- b) Drive storage capacity – full size of all chips transferred in LBA
- c) Max LBA – the number of LBA, set aside for user data (drive storage capacity set in passport)
- d) Logical (direct) data reading shows the possibility of building translator tables by utility. If it is possible, utility will allow user to make sector by sector copy from drive bypassing any SSD preparations (like interleave, page transformation, joining by bytes, etc), and save whole SSD image.

3. Utility features

Tests:

- a) *Utility status -> Reading parameters*: here you can chose parameters for memory chip reading (interleave elimination, Interleave type, XOR elimination, etc).

- b) *Service information -> security subsystem -> viewing of Passwords info*: possible master and user passwords in binary view will be shown in log. In some cases, this test can take a very long time. User can interrupt this test anytime, then a list of possible passwords will be shown in the log.
- c) *Service information -> BAD block table*: The list of BAD blocks for each NAND chip will be shown in log window.
- d) *User data -> Dumps reading*: Saving dumps from memory chip into selected directory.
- e) *Drive testing -> SSD reinitialization*: getting information about drive and attempt for translator tables building.
- f) *Drive testing -> NAND flash memory chip testing*: utility runs a test reading of the specified range of selected chips. Possible messages:
 - ECC error – impossible to correct bit errors
 - CRC error – checksum mismatching
 - Bit errors – read page contains bit errors
 - Data corrupted (ECC&CRC errors) - failed to correct the bit and checksum errors at the same time when reading
 - BAD block (low-high) – page situated within BAD block
 - Reading error
 - Page fill by 0xFF

Utility extensions.

- a) *Tools -> Utility extensions -> Dump viewing*: direct memory chip reading or reading logical image (if it is possible).