10s: Comparison & Bitwise Logic Operations

Value	Mnemonic	δ	α	Description
0x10	LT	2	1	Less-than comparison. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] < \boldsymbol{\mu}_{\mathbf{s}}[1] \\ 0 & \text{otherwise} \end{cases}$
0x11	GT	2	1	Greater-than comparison. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] > \boldsymbol{\mu}_{\mathbf{s}}[1] \\ 0 & \text{otherwise} \end{cases}$
0x12	SLT	2	1	Signed less-than comparison. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] < \boldsymbol{\mu}_{\mathbf{s}}[1] \\ 0 & \text{otherwise} \end{cases}$ Where all values are treated as two's complement signed 256-bit integers.
0x13	SGT	2	1	Signed greater-than comparison. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] > \boldsymbol{\mu}_{\mathbf{s}}[1] \\ 0 & \text{otherwise} \end{cases}$ Where all values are treated as two's complement signed 256-bit integers.
0x14	EQ	2	1	Equality comparison. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] = \boldsymbol{\mu}_{\mathbf{s}}[1] \\ 0 & \text{otherwise} \end{cases}$
0x15	ISZERO	1	1	Simple not operator. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0] = 0 \\ 0 & \text{otherwise} \end{cases}$
0x16	AND	2	1	Bitwise AND operation. $\forall i \in [0..255] : \boldsymbol{\mu}'_{\mathbf{s}}[0]_{\text{i}} \equiv \boldsymbol{\mu}_{\mathbf{s}}[0]_{\text{i}} \wedge \boldsymbol{\mu}_{\mathbf{s}}[1]_{\text{i}}$
0x17	OR	2	1	Bitwise OR operation. $\forall i \in [0..255] : \boldsymbol{\mu}'_{\mathbf{s}}[0]_{\text{i}} \equiv \boldsymbol{\mu}_{\mathbf{s}}[0]_{\text{i}} \vee \boldsymbol{\mu}_{\mathbf{s}}[1]_{\text{i}}$
0x18	XOR	2	1	Bitwise XOR operation. $\forall i \in [0..255] : \boldsymbol{\mu}'_{\mathbf{s}}[0]_{\text{i}} \equiv \boldsymbol{\mu}_{\mathbf{s}}[0]_{\text{i}} \oplus \boldsymbol{\mu}_{\mathbf{s}}[1]_{\text{i}}$
0x19	NOT	1	1	Bitwise NOT operation. $\forall i \in [0..255] : \boldsymbol{\mu}'_{\mathbf{s}}[0]_{\text{i}} \equiv \begin{cases} 1 & \text{if} \quad \boldsymbol{\mu}_{\mathbf{s}}[0]_{\text{i}} = 0 \\ 0 & \text{otherwise} \end{cases}$
0x1a	BYTE	2	1	Retrieve single byte from word. $\forall i \in [0..255] : \boldsymbol{\mu}'_{\mathbf{s}}[0]_{\text{i}} \equiv \begin{cases} \boldsymbol{\mu}_{\mathbf{s}}[1]_{(i-248+8\boldsymbol{\mu}_{\mathbf{s}}[0])} & \text{if} \quad i \geq 248 \wedge \boldsymbol{\mu}_{\mathbf{s}}[0] < 32 \\ 0 & \text{otherwise} \end{cases}$ For the Nth byte, we count from the left (i.e. N=0 would be the most significant in big endian).
0x1b	SHL	2	1	Left shift operation. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv (\boldsymbol{\mu}_{\mathbf{s}}[1] \times 2^{\boldsymbol{\mu}_{\mathbf{s}}[0]}) \bmod 2^{256}$
0x1c	SHR	2	1	Logical right shift operation. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \lfloor \boldsymbol{\mu}_{\mathbf{s}}[1] \div 2^{\boldsymbol{\mu}_{\mathbf{s}}[0]} \rfloor$
0x1d	SAR	2	1	Arithmetic (signed) right shift operation. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \lfloor \boldsymbol{\mu}_{\mathbf{s}}[1] \div 2^{\boldsymbol{\mu}_{\mathbf{s}}[0]} \rfloor$ Where $\boldsymbol{\mu}'_{\mathbf{s}}[0]$ and $\boldsymbol{\mu}_{\mathbf{s}}[1]$ are treated as two's complement signed 256-bit integers, while $\boldsymbol{\mu}_{\mathbf{s}}[0]$ is treated as unsigned.

20s: KECCAK256

Value	Mnemonic	δ	α	Description
0x20	KECCAK256	2	1	Compute Keccak-256 hash. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \texttt{KEC}(\boldsymbol{\mu}_{\mathbf{m}}[\boldsymbol{\mu}_{\mathbf{s}}[0] \ldots (\boldsymbol{\mu}_{\mathbf{s}}[0] + \boldsymbol{\mu}_{\mathbf{s}}[1] - 1)])$ $\boldsymbol{\mu}'_{\text{i}} \equiv M(\boldsymbol{\mu}_{\text{i}}, \boldsymbol{\mu}_{\mathbf{s}}[0], \boldsymbol{\mu}_{\mathbf{s}}[1])$