C6.2.5 ADD (shifted register)

Add (shifted register) adds a register value and an optionally-shifted register value, and writes the result to the destination register.

32-bit variant

Applies when \(sf == 0\).

\[\text{ADD} \ <Wd>, <Wn>, <Wm>\{, <shift> \#<amount>\}\]

64-bit variant

Applies when \(sf == 1\).

\[\text{ADD} \ <Xd>, <Xn>, <Xm>\{, <shift> \#<amount>\}\]

Decode for all variants of this encoding

\[
\begin{align*}
\text{integer } d &= \text{UInt}(Rd); \\
\text{integer } n &= \text{UInt}(Rn); \\
\text{integer } m &= \text{UInt}(Rm); \\
\text{integer } \text{datasize} &= \text{if } sf == '1' \text{ then } 64 \text{ else } 32; \\
\text{if } \text{shift} == '11' \text{ then } \text{ReservedValue}(); \\
\text{if } sf == '0' \&\& \text{imm6}<5> == '1' \text{ then } \text{ReservedValue}(); \\
\text{ShiftType } \text{shift_type} &= \text{DecodeShift}(\text{shift}); \\
\text{integer } \text{shift_amount} &= \text{UInt}(\text{imm6});
\end{align*}
\]

Assembler symbols

\(<Wd>\) Is the 32-bit name of the general-purpose destination register, encoded in the "Rd" field.

\(<Wn>\) Is the 32-bit name of the first general-purpose source register, encoded in the "Rn" field.

\(<Wm>\) Is the 32-bit name of the second general-purpose source register, encoded in the "Rm" field.

\(<Xd>\) Is the 64-bit name of the general-purpose destination register, encoded in the "Rd" field.

\(<Xn>\) Is the 64-bit name of the first general-purpose source register, encoded in the "Rn" field.

\(< Xm>\) Is the 64-bit name of the second general-purpose source register, encoded in the "Rm" field.

\(<\text{shift}>\) Is the optional shift type to be applied to the second source operand, defaulting to LSL and encoded in the "shift" field. It can have the following values:

- LSL when \(\text{shift} = 00\)
- LSR when \(\text{shift} = 01\)
- ASR when \(\text{shift} = 10\)

The encoding \(\text{shift} = 11\) is reserved.

\(<\text{amount}>\) For the 32-bit variant: is the shift amount, in the range 0 to 31, defaulting to 0 and encoded in the "imm6" field.

For the 64-bit variant: is the shift amount, in the range 0 to 63, defaulting to 0 and encoded in the "imm6" field.
Operation

\[
\begin{align*}
\text{bits(datasize)} & \text{ result;} \\
\text{bits(datasize} & \text{ operand1} = X[n]; \\
\text{bits(datasize} & \text{ operand2} = \text{ShiftReg}(m, \text{shift_type}, \text{shift_amount}); \\
\text{(result, -)} & = \text{AddWithCarry}(\text{operand1, operand2, '0'}); \\
X[d] & = \text{result};
\end{align*}
\]