ARM v8 architecture Introduction

A1.5 Advanced SIMD and floating-point support

A1.6 The ARM memory model

A1.7 ARMv8 architecture extension

A1.5 Advanced SIMD and floating-point support

AArch32	AArch64
General registers -> SIMD instructions	X
SIM & FP registers -> advanced SIMD instructions	SIM & FP registers -> advanced SIMD instructions

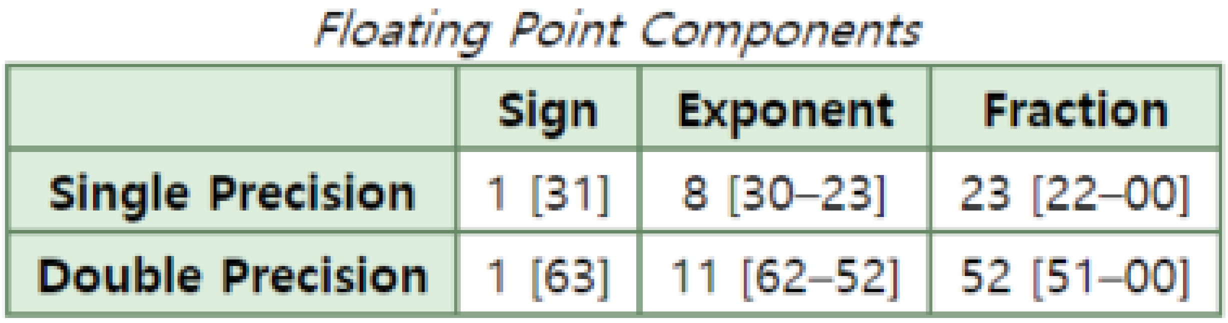
ARMv8 은 single-precision(32 bits) 와 double-precision(64 bits)를 지원함 (IEEE 754).

half-precision(16 bits)은 single과 double을 conversions해서 사용하도록 지원함(for data storage).

A1.5 Advanced SIMD and floating-point support

SIMD instruction 은 다음을 지원한다.

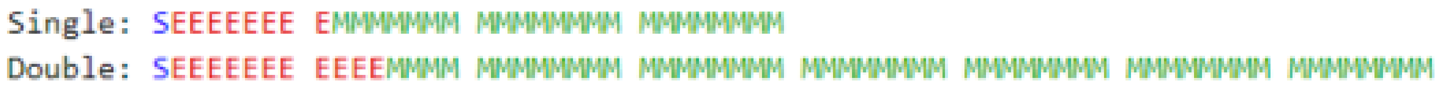
1. AArch64 : single-precision and double-precision
2. AArch32 : single-precision
3. AArch64& AArch32 : ARMv8.2-FP16이 실행되면, half -precision



Floating Point Components

	Sign	Exponent	Fraction
Single Precision	1 [31]	8 [30–23]	23 [22–00]
Double Precision	1 [63]	11 [62–52]	52 [51–00]

[Floating point components]
Sign bit
Exponent
Mantissa(fraction mantissa, 가수)



```
Single: SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM
Double: SEEEEEEE EEEEMMMM MMMMMMMM MMMMMMMM MMMMMMMM MMMMMMMM MMMMMMMM MMMMMMMM
```

http://steve.hollasch.net/cgindex/coding/ieeefloat.html

A1.5 Advanced SIMD and floating-point support

Floating-point support (in AArch64 state SIMD)는 다음을 준수한다.

1. Configurable rounding modes
2. Configurable default NaN behavior
3. Configurable Flush-to-zero behavior

Floating-point support (in AArch32 state SIMD)는 다음을 준수한다.

1. Denormalized numbers are flushed to zero
2. Default NaNs
3. Round to Nearest rounding mode
4. Untrapped floating-point exception handling is used for all floating-point exceptions

A1.5 Advanced SIMD and floating-point support

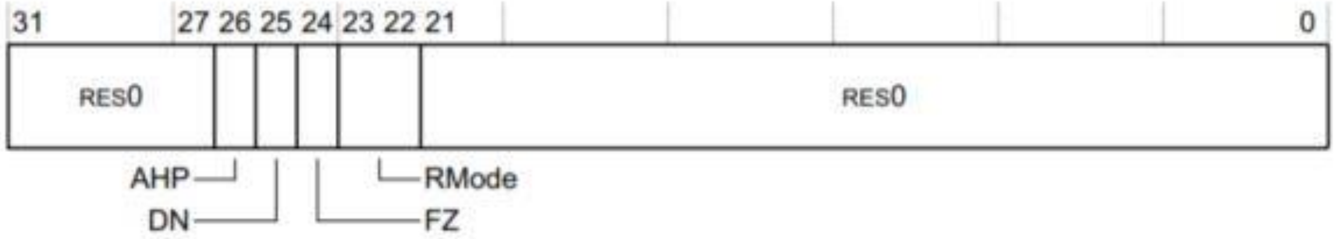
Registers control floating-point operation & return floating-point status information	

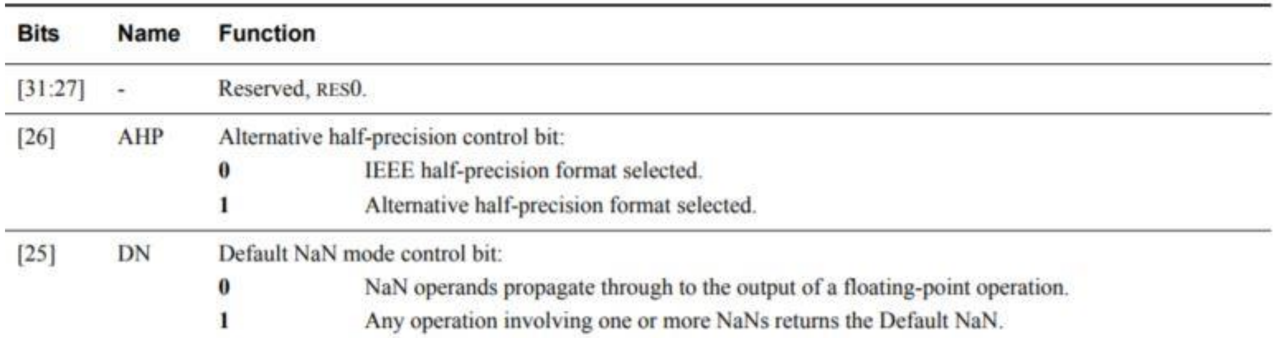
AArch64	FPCR(floating-point control registers)
	FPSR(floating-point status registers)

AArch32	FPSCR(FPCR+FPSR/ floating-point status control registers)

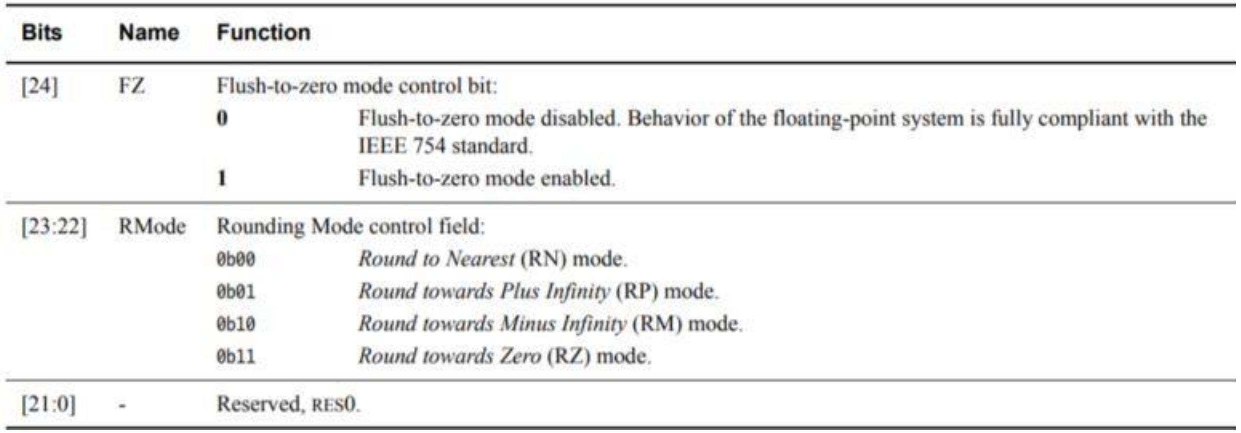
A1.5 Advanced SIMD and floating-point support

AArch64 : FPCR(floating-point control registers)





Bits	Name	Function
[31:27]	-	Reserved, RES0.
[26]	AHP	Alternative half-precision control bit: **0** IEEE half-precision format selected. **1** Alternative half-precision format selected.
[25]	DN	Default NaN mode control bit: **0** NaN operands propagate through to the output of a floating-point operation. **1** Any operation involving one or more NaNs returns the Default NaN.

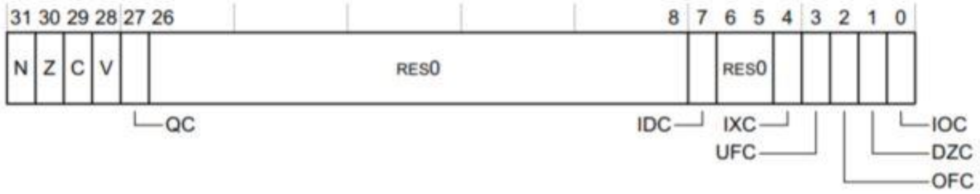


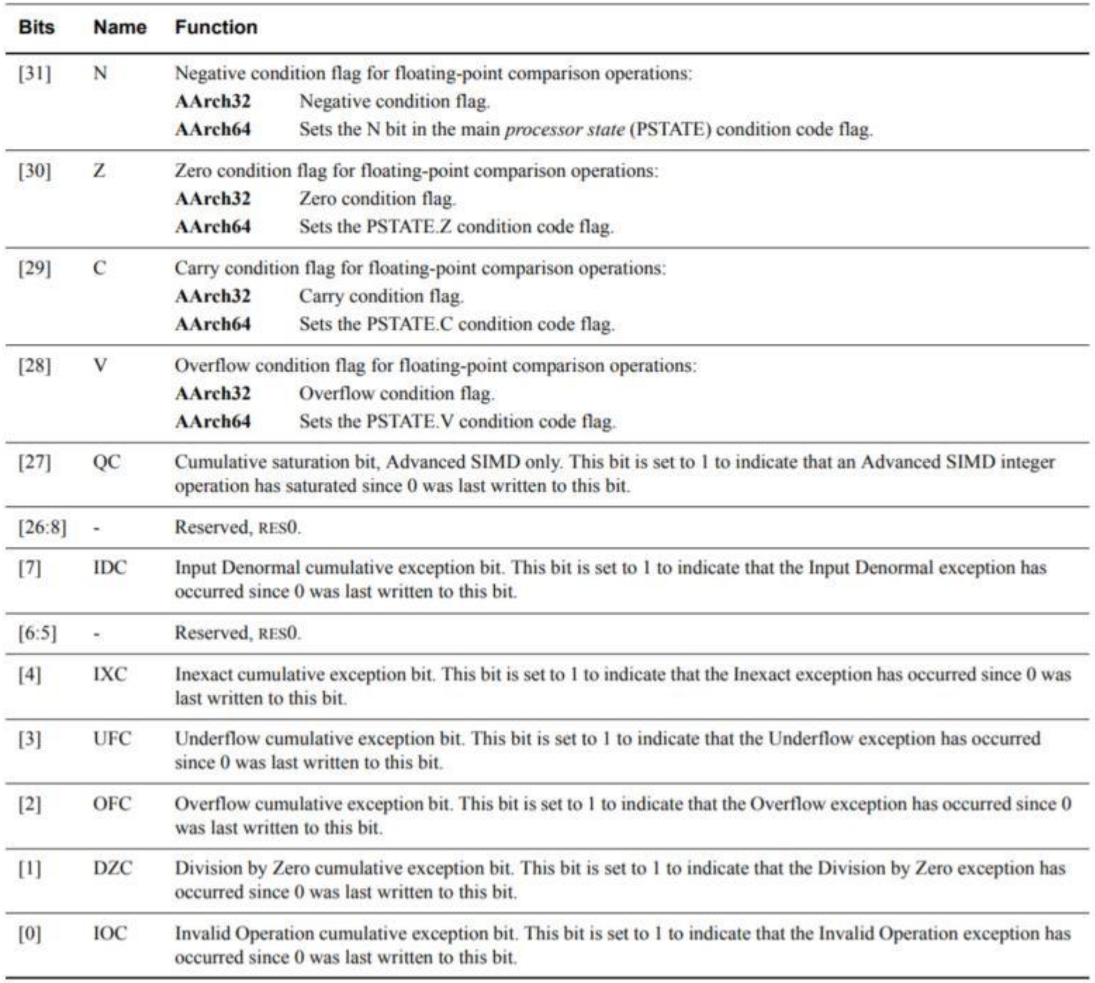
Bits	Name	Function
[24]	FZ	Flush-to-zero mode control bit: **0** Flush-to-zero mode disabled. Behavior of the floating-point system is fully compliant with the IEEE 754 standard. **1** Flush-to-zero mode enabled.
[23:22]	RMode	Rounding Mode control field: `0b00` *Round to Nearest* (RN) mode. `0b01` *Round towards Plus Infinity* (RP) mode. `0b10` *Round towards Minus Infinity* (RM) mode. `0b11` *Round towards Zero* (RZ) mode.
[21:0]	-	Reserved, RES0.

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0488d/CIHCACFF.html

A1.5 Advanced SIMD and floating-point support

AArch64 : FPSR(floating-point status registers)



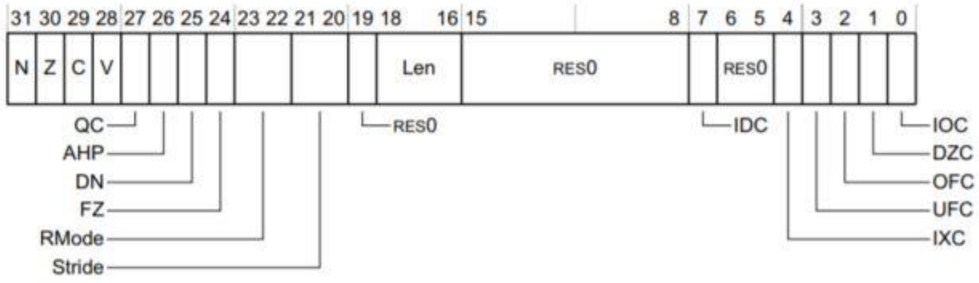


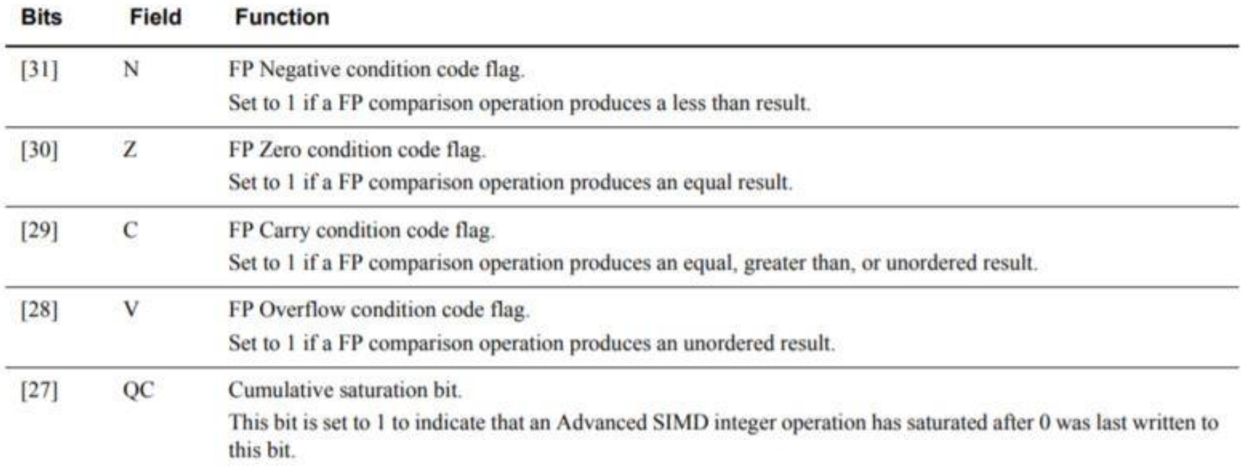
Bits	Name	Function
[31]	N	Negative condition flag for floating-point comparison operations: **AArch32** Negative condition flag. **AArch64** Sets the N bit in the main *processor state* (PSTATE) condition code flag.
[30]	Z	Zero condition flag for floating-point comparison operations: **AArch32** Zero condition flag. **AArch64** Sets the PSTATE.Z condition code flag.
[29]	C	Carry condition flag for floating-point comparison operations: **AArch32** Carry condition flag. **AArch64** Sets the PSTATE.C condition code flag.
[28]	V	Overflow condition flag for floating-point comparison operations: **AArch32** Overflow condition flag. **AArch64** Sets the PSTATE.V condition code flag.
[27]	QC	Cumulative saturation bit, Advanced SIMD only. This bit is set to 1 to indicate that an Advanced SIMD integer operation has saturated since 0 was last written to this bit.
[26:8]	-	Reserved, RES0.
[7]	IDC	Input Denormal cumulative exception bit. This bit is set to 1 to indicate that the Input Denormal exception has occurred since 0 was last written to this bit.
[6:5]	-	Reserved, RES0.
[4]	IXC	Inexact cumulative exception bit. This bit is set to 1 to indicate that the Inexact exception has occurred since 0 was last written to this bit.
[3]	UFC	Underflow cumulative exception bit. This bit is set to 1 to indicate that the Underflow exception has occurred since 0 was last written to this bit.
[2]	OFC	Overflow cumulative exception bit. This bit is set to 1 to indicate that the Overflow exception has occurred since 0 was last written to this bit.
[1]	DZC	Division by Zero cumulative exception bit. This bit is set to 1 to indicate that the Division by Zero exception has occurred since 0 was last written to this bit.
[0]	IOC	Invalid Operation cumulative exception bit. This bit is set to 1 to indicate that the Invalid Operation exception has occurred since 0 was last written to this bit.

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0488d/CIHCACFF.html

A1.5 Advanced SIMD and floating-point support

AArch32 : FPSCR(floating-point status control registers)





Bits	Field	Function
[31]	N	FP Negative condition code flag. Set to 1 if a FP comparison operation produces a less than result.
[30]	Z	FP Zero condition code flag. Set to 1 if a FP comparison operation produces an equal result.
[29]	C	FP Carry condition code flag. Set to 1 if a FP comparison operation produces an equal, greater than, or unordered result.
[28]	V	FP Overflow condition code flag. Set to 1 if a FP comparison operation produces an unordered result.
[27]	QC	Cumulative saturation bit. This bit is set to 1 to indicate that an Advanced SIMD integer operation has saturated after 0 was last written to this bit.

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0488d/CIHCACFF.html

A1.5.1 instruction support

The Advanced SIMD &FP instructions support :

- load&store for single elements and vectors of multiple elements
  - (single elements는 scalar elements 라고도 한다.)
- Data processing on single and multiple elements for both integer and FP data types
- when CompNum is implemented, complex number arithmetic
- FP conversion between different levels of precision
- Conversion between FP, fixed-point integer, and integer data type
- Floating point rounding

A1.5.3 ARM standard floating-point input and output values

Input	Output
Zeros	Zeros
Normalized numbers	Normalized numbers
Denormalized numbers are flushed to 0 Before floating-point operations	Results that are less than the minimum normalized number are flushed to zero
NaNs	NaNs produced in floating-point operations are always the default NaN
Infinities	Infinities

A1.5.3 ARM standard floating-point input and output values

[exponent 가 all 1 이거나 0 인 경우]

Zero	exponent 와 fraction 모두 0 (단, -0 , +0 이 존재한다.)
Denormalized value	exponent 가 모두 0 이고 fraction은 non-zero Single : $(-1)^s \times 0.f \times 2^{-126}$ Double : $(-1)^s \times 0.f \times 2^{-1022}$
Infinity	exponent 와 fraction이 모두 1(+∞)이거나 0(-∞)
NaN (Not a Number)	real number가 아닌 것. 연산과정에서 잘못된 입력을 받아 계산을 하지 못했음을 나타내는 기호 Exponent가 모두 1이고 fraction 은 non-zero

A1.5.4 Flush-to-zero

floating-point exception 은 다음과 같을 때 발생한다

- Invalid operation
- Division by zero
- Overflow
- Underflow
- Inexact calculation

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/com.ibm.aix.genprogc/floating-point_except.htm

A1.5.4 Flush-to-zero

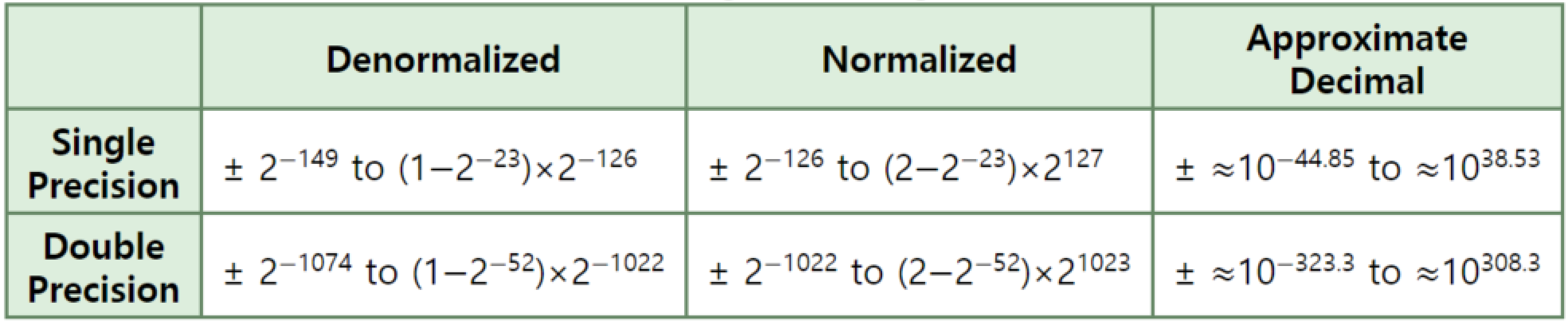
floating-point exception 은 다음과 같을 때 발생한다

- Invalid operation (잘못된 연산)
  연산시에 잘못된 인자가 사용되는 경우에 발생하며 기본적으로는 NaN을 반환한다.
- Division by zero (0으로 나누기)
  피 연산자가 0인 경우에 발생하며 기본적으로는 ±∞를 반환한다.
- Overflow
  반올림 처리 후 표현범위를 벗어날 때 발생하며 기본적으로는 ±∞를 반환한다.
- Underflow
  반올림 처리 후 표현범위보다 너무 작은 값이 될 때 발생하며 기본적으로는 denormalized number을 반환한다. 보통 inexact(부정확함)과 같이 발생된다.
- Inexact (부정확함)
  연산의 결과가 부정확할 때 발생한다. 부정확하다는 것은 반올림 처리시 0이 아닌 수가 잘리는 상황이 발생할 때가 대부분이다. 기본적으로는 반올림 처리된 결과를 반환한다.

https://nybounce.wordpress.com/2016/06/24/ieee-754-floating-point%EB%B6%80%EB%8F%99%EC%86%8C%EC%88%98%EC%A0%90-%EC%82%B0%EC%88%A0%EC%97%90-%EB%8C%80%ED%95%98%EC%97%AC/

A1.5.4 Flush-to-zero

Floating Point Range



	Denormalized	Normalized	Approximate Decimal
Single Precision	$\pm\ 2^{-149}$ to $(1-2^{-23})\times 2^{-126}$	$\pm\ 2^{-126}$ to $(2-2^{-23})\times 2^{127}$	$\pm\ \approx 10^{-44.85}$ to $\approx 10^{38.53}$
Double Precision	$\pm\ 2^{-1074}$ to $(1-2^{-52})\times 2^{-1022}$	$\pm\ 2^{-1022}$ to $(2-2^{-52})\times 2^{1023}$	$\pm\ \approx 10^{-323.3}$ to $\approx 10^{308.3}$

- Negative numbers less than $-(2-2^{-23}) \times 2^{127}$ (*negative overflow*)
- Negative numbers greater than -2^{-149} (*negative underflow*)
- Zero
- Positive numbers less than 2^{-149} (*positive underflow*)
- Positive numbers greater than $(2-2^{-23}) \times 2^{127}$ (*positive overflow*)

A1.5.4 Flush-to-zero

Flush to zero mode

1. Input denormal floating-point exception occurs only in flush-to-0 mode

The occurrence of all floating-point exceptions except Input Denormal is determined using the input values after flush-to-zero processing has occurred.

A1.5.4 Flush-to-zero

Flush to zero mode

2. The result of a floating-point operation is flushed to zero if the result of the operation before rounding satisfies the condition :

0 < Abs(result) < MinNorm, where:

MinNorm is 2–14 for half-precision.

MinNorm is2-126 for single-precision.

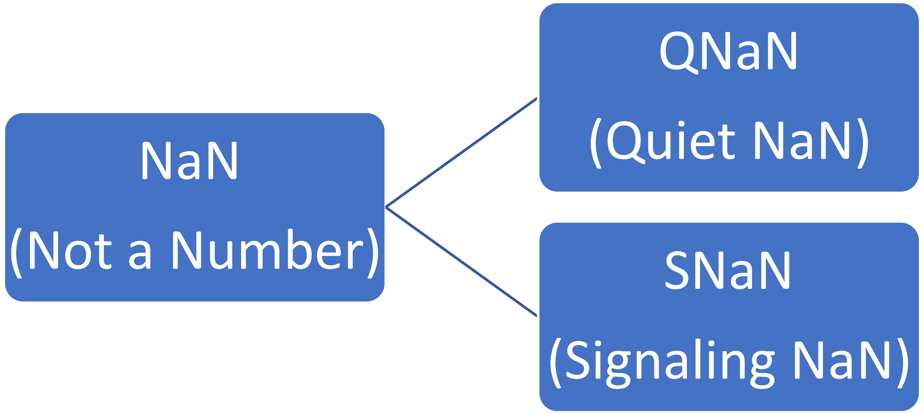
MinNorm is2-1022 for double-precision.

A1.5.4 Flush-to-zero

Flush to zero mode

3. An Inexact floating-point exception does not occur if the result is flushed to zero, even though the final result of zero is not equivalent to the value that would be produced if the operation were performed with unbounded precision and exponent range.

A1.5.4 NaN handling and the Default NaN



Invalid operation (잘못된 연산) (of floating-point exception)
연산시에 잘못된 인자가 사용되는 경우에 발생하며
기본적으로는 NaN을 반환한다.

QNaN : 결과가 수학적으로 정의되지 않은 경우 작업에서 생성

SNaN : 연산에 사용될 때 예외를 알리는 데 사용

Default NaN : Default NaN 모드(FPSCR[DN]) == 1이면 Default NaN mode)에서
입력 NaN를 포함한 오퍼레이션의 결과, 또는 NaN 결과를 생성 한 오퍼레이션
Default의 NaN를 돌려줌

A1.5.4 NaN handling and the Default NaN

1. 연산이 Invalid operation floating-point exception 을 발생시키면 result = QNaN
2. 피연산자에 QNaN(SNaN은 안됨)이 있으면 result = input NaN

Default NaN mode == disable일 때

A. untrapped invalid operation floating-point exception이 발생했을 때

QNaN result : 첫번째 SNaN 피연산자 (exception이 발생함 = operand는 최소 1개이상의 SNaN을 가짐)

QNaN result : 그 외에는 Default NaN

B. untrapped invalid operation floating-point exception이 발생하지 않았을 때

연산자 중 최소 하나이상이 QNaN 연산자이면

QNaN result = 첫번째 QNaN 연산자

Default NaN mode==enable 일 때

A. untrapped invalid operation floating-point exception이 발생했을 때

B. 1개 이상의 QNaN input을 가질 때 (SNaN은 안됨)

A,B 둘다 result = Default NaN.

A1.5.4 NaN handling and the Default NaN

Default NaN

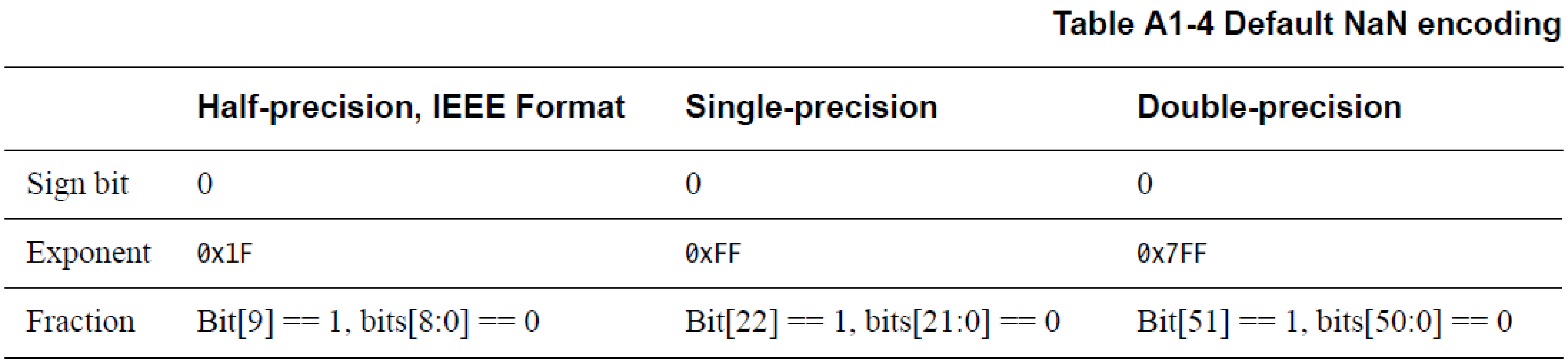


Table A1-4 Default NaN encoding

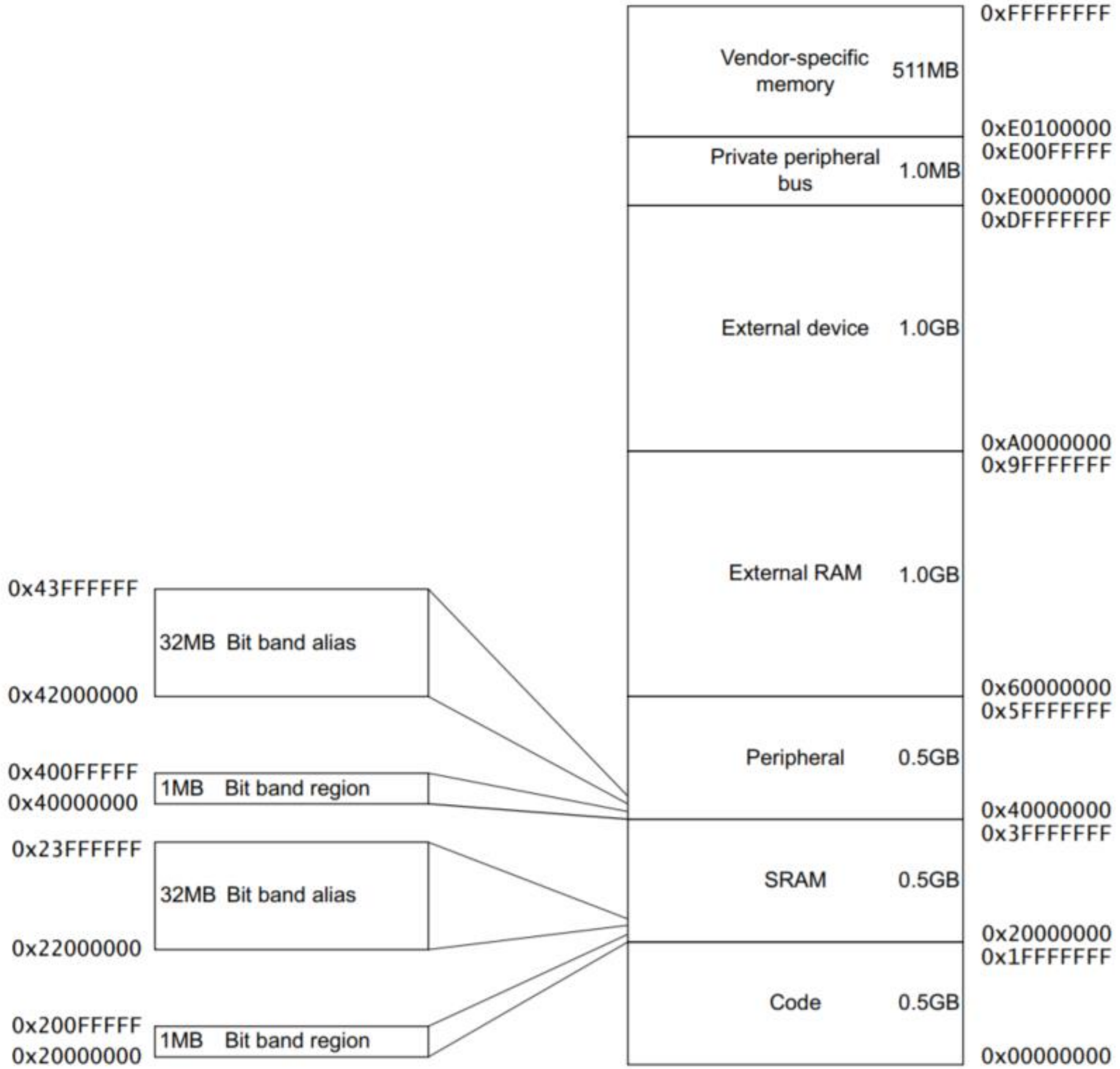
	Half-precision, IEEE Format	Single-precision	Double-precision
Sign bit	0	0	0
Exponent	0x1F	0xFF	0x7FF
Fraction	Bit[9] == 1, bits[8:0] == 0	Bit[22] == 1, bits[21:0] == 0	Bit[51] == 1, bits[50:0] == 0

A1.6 The ARM memory model

The ARM memory model supports:

- Generating an exception on an unaligned memory access.
- Restricting access by applications to specified areas of memory.
- Translating *virtual addresses* (VAs) provided by executing instructions to *physical addresses* (PAs).
- Altering the interpretation of multi-byte data between big-endian and little-endian.
- Controlling the order of accesses to memory.
- Controlling caches and address translation structures.
- Synchronizing access to shared memory by multiple PEs.

A1.6 The ARM memory model


0xFFFFFFFF
Vendor-specific memory
511MB
0xE0100000
0xE00FFFFF
Private peripheral bus
1.0MB
0xE0000000
0xDFFFFFFF
External device
1.0GB
0xA0000000
0x9FFFFFFF
External RAM
1.0GB
0x60000000
0x5FFFFFFF
Peripheral
0.5GB
0x40000000
0x3FFFFFFF
SRAM
0.5GB
0x20000000
0x1FFFFFFF
Code
0.5GB
0x00000000
0x43FFFFFF
32MB Bit band alias
0x42000000
0x400FFFFF
1MB Bit band region
0x40000000
0x23FFFFFF
32MB Bit band alias
0x22000000
0x200FFFFF
1MB Bit band region
0x20000000