Base
Instructions

SIMD&FP
Instructions

SVE
Instructions

SME
Instructions

Index by
Encoding

Sh
Pseuc

LDURB

Load Register Byte (unscaled) calculates an address from a base register and an immediate offset, loads a byte from memory, zero-extends it, and writes it to a register. For information about memory accesses, see *Load/ Store addressing modes*.

31 30	29 28 27	26	25 24 23	22 21	20	19 18 17 16 15 14 13 12	11 10	9 8 7 6 5	4 3 2 1 0
0 0	1 1 1	0	0 0 0	0 1	0	imm9	0 0	Rn	Rt
size			opc						

>   LDURB **<Wt>, [<Xn|SP>{, #<simm>}]**

```
bits(64) offset = SignExtend(imm9, 64);
```

Assembler Symbols

<Wt>   Is the 32-bit name of the general-purpose register to be transferred, encoded in the "Rt" field.

<Xn|SP>   Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field.

<simm>   Is the optional signed immediate byte offset, in the range -256 to 255, defaulting to 0 and encoded in the "imm9" field.

Shared Decode

```
integer n = UInt(Rn);
integer t = UInt(Rt);

boolean tagchecked = n != 31;
```

Operation

```
bits(64) address;
bits(8) data;

boolean privileged = PSTATE.EL != EL0;
AccessDescriptor accdesc = CreateAccDescGPR(MemOp_LOAD, FALSE, privileg

if n == 31 then
    CheckSPAlignment();
    address = SP[];
else
    address = X[n, 64];

address = address + offset;

data = Mem[address, 1, accdesc];
X[t, 32] = ZeroExtend(data, 32);
```

Operational information

If PSTATE.DIT is 1, the timing of this instruction is insensitive to the value of the data being loaded or stored.

Base Instructions	SIMD&FP Instructions	SVE Instructions	SME Instructions	Index by Encoding	Sh Pseud