

# Princeton University

## COS 217: Introduction to Programming Systems

### ARMv8 Memory Operands

Let `reg[Xn]` mean "the contents of register *Xn*".

#### **Register Memory Operands**

`[Xn]` Compute an address as `reg[Xn]`.

Examples:

- (1) `ldr x1, [x0]`
- (2) `str x1, [x0]`
- (3) `ldr w1, [x0]`
- (4) `str w1, [x0]`

#### **Immediate Offset Memory Operands**

`[Xn, imm]` Compute an address as `reg[Xn] + imm`.

Examples:

- (5) `ldr x1, [x0, 4]`
- (6) `str x1, [x0, 8]`
- (7) `ldr w1, [x0, 12]`
- (8) `str w1, [x0, 16]`

#### **Register Offset Memory Operands**

`[Xn, Xm]` Compute an address as `reg[Xn] + reg[Xm]`.

Examples:

- (9) `ldr x2, [x0, x1]`
- (10) `str x2, [x0, x1]`
- (11) `ldr w2, [x0, x1]`
- (12) `str w2, [x0, x1]`

#### **Scaled Register Offset Memory Operands**

`[Xn, Xm, lsl 3]` Compute an address as `reg[Xn] + (reg[Xm] << 3)`.  
The loaded/stored object must consist of 8 bytes.

`[Xn, Xm, lsl 2]` Compute an address as `reg[Xn] + (reg[Xm] << 2)`.  
The loaded/stored object must consist of 4 bytes.

`[Xn, Xm, lsl 1]` Compute an address as `reg[Xn] + (reg[Xm] << 1)`.  
The loaded/stored object must consist of 2 bytes.

Examples:

- (13) `ldr x2, [x0, x1, lsl 3]`
- (14) `str x2, [x0, x1, lsl 3]`
- (15) `ldr w2, [x0, x1, lsl 2]`
- (16) `str w2, [x0, x1, lsl 2]`
- (17) `ldrh w2, [x0, x1, lsl 1]`
- (18) `strh w2, [x0, x1, lsl 1]`