Consider our ISA x295++ with the following components:

- Memory model: \( 2^m \times n = 2^{12} \times 16 \)
- Word size: 16 bits
- 8 \times 16-bit registers \( \rightarrow (r0 \leftrightarrow r7) \) each uniquely specified using 3 bits
- Base + Displacement mode
- Template: 
  
<table>
<thead>
<tr>
<th>Opcode</th>
<th>Dest</th>
<th>Src</th>
<th>Src (value)</th>
</tr>
</thead>
</table>

  If I use the LOAD instruction \( \text{LOAD value}(rA), \ rC \) in my assembly code to step through an array \( A \) of \( \text{short} \) (\( rA \) is the address of the first element of array \( A \)) and load each of these elements the second element into \( rC \) in turn, to which \( \text{value} \) shall I set \( \text{value} \)?

  Answer:
  - The instruction \( \text{LOAD value}(rA), \ rC \) becomes \( \text{LOAD} \ 1(r0), \ r1 \) in my assembly code
  - As you can see, \( \text{value} \) is set to 1 because

    - array \( A \) is an array of \( \text{short} \) data type elements each has a size of 16 bits:
      
      | First element of \( A \) | Second element of \( A \) | Third element of \( A \) | ... | Last element of \( A \) |
      |-------------------------|-------------------------|-------------------------|-----|-------------------------|
      | \( A \)                 | \( A + 1 \)             | \( A + 2 \)             |     | \( A + \text{arraySize} -1 \) |

    - and since this ISA is a word-addressable memory model, i.e., every 16-bit chunk \( (2^{12} \times 16) \) is given a memory address, then element 0 of array \( A \) is given an address (here \( A \)) and element 1 of array \( A \), located 16 bits further, is also given a memory address (here \( A+1 \))
    - So, to go from the first element of array \( A \) to the next requires us to add 1 to the memory address \( A \)
If you answered 2 (meaning: 2 bytes), you are on the right track!

- You need to express what 2 bytes is in this new ISA x295++, i.e., a “chunk of memory that has its own address"

If you answered 16 (meaning: 16 bits), you are on the right track!

- You need to express what 16 bits is in this new ISA x295++, i.e., a “chunk of memory that has its own address"

So, the bottom line is, value must have the same “unit” as the content of rA i.e., memory address

- In this ISA, we could name a “chunk of memory that has its own address”, i.e., the “unit” of the memory address, we could name it doublebyte

- So, the displacement value would have the “unit” doublebyte

- By adding 1 to the memory address, this 1 is understood as 1 doublebyte and we get the memory address of the next addressable chunk of memory 16 bits (or 2 bytes) further, i.e., 1 doublebyte of memory further

- If we add 2 to the memory address, this 2 would be understood as 2 doublebytes and we would get the memory address of the addressable chunk of memory 32 bits (or 4 bytes) further, i.e., 2 doublebytes of memory further
  - In our example, we would get the address of the third element of A

- If we add 16 to the memory address, this 16 would be understood as 16 doublebytes and we would get the memory address of the addressable chunk of memory 256 bits (or 32 bytes) further, i.e., 16 doublebytes of memory further

I hope the above makes sense! If it does not, please, feel free to ask us questions!

You may also review the Example of Memory Models posted on Friday March 6 Lecture 22