Problems With Basic Design
(Figure 6.13, p. 243)

- The following registers must have an **INCREMENT** control line: *AR* and *PC*.

- The following registers must have a **LOAD** control line: *AR*, *PC*, *DR*, *IR*, *R* and *TR*.

- The *AR* and *IR* registers **do not supply data** to the other registers or memory.

- The entire bus capacity (all 16 bits) **is not fully used** by all registers.

- Data routing collisions occur! For example:
  - During *T*₂ of the FETCH cycle, *IR ← DR* and *AR ← PC* must be done simultaneously.
  - During *T*₄ in an *LDAC* instruction (and in other times), *TR ← DR* and *DR ← M* must be done simultaneously.
  - During *T*₅ in an *LDAC* instruction (and in other times), *DR* and *TR* must be put on the bus simultaneously (*AR ← DR | TR*).

- *D*[7..0] need to be **bidirectional** (or else data cannot be written to the memory!).

- Where is the *Z* register?

- What about the ALU interface and controls?
Some Solutions to Routing Problems of Basic Design

- Make \( AR \) and \( PC \) counters with load capability.

- Make \( DR \), \( IR \), \( R \) and \( TR \) data registers with load capability.

- Remove the connections from the \( AR \) and \( IR \) registers to the bus.

- Connect 8-bit registers to bits 7..0 of the bus.

- To resolve data collisions:
  
  - Since \( IR \) receives data from the \( DR \) only, construct a direct path from the \( DR \) to the \( IR \).
  
  - Since \( TR \) receives data from the \( DR \) only, construct a direct path from the \( DR \) to the \( TR \). Remove the connection from \( TR \) to the bus (since it is no longer needed).

  - Route the output of \( DR \) to both sets of bits 7..0 and 15..8 of the bus. Separate buffers with different enable signals must be used because \( DR \) should not be active on both halves of the bus simultaneously.

- Use buffer gates to make \( D[7..0] \) bidirectional.

- Use a NOR gate with the output of the ALU and connect the result to the \( Z \) register.

- ALU interface delayed until later.