We require a hardware accelerator to compute

\[ f(x) = \sum_{i=0}^{7} C[i] \times D[i + x] \]

where all values are 20-bit signed integers and the array \( C[i] \) contains design-time constants. The array \( D[] \) will contain 1024 values. We need to evaluate \( f(x) \) as fast as possible. The values of \( x \) are unpredictable and one word of \( D \) gets updated by a separate process after every 50 or so evaluations of \( f \). Ignore overflow.

\( (a) \) An early design holds both \( C \) and \( D \) in a common, single-ported RAM memory (i.e. one with one address bus). Given that single-cycle multiply-accumulate blocks are available, give a rough estimate of the performance of our system in clocks per evaluation. [3 marks]

\( (b) \) Suggest a small change to the early design that improves its performance and estimate the resulting performance. Suggest one or more further sensible improvements and indicate the new performance. [6 marks]

\( (c) \) The output from a High-Level Synthesis (HLS) compiler is generally RTL which is then fed to a logic synthesis compiler. Identify four tasks performed by the combined flow, explaining which stage does which. [4 marks]

\( (d) \) Using a block-structured high-level language like C or Java, or using pseudocode, briefly write an implementation of \( f(x) \) that would be amenable to HLS compilation. State three properties of your implementation that make it likely to be acceptable to an HLS compiler and/or give good performance. What might influence the choice of design from the solution space in part \( (b) \)? You may ignore the mechanism by which \( D \) is updated. [7 marks]