QUESTION 1:

An operation will be applied to each element in an array. A hardware pipeline of five segments using combinatorial circuitry has been designed for this operation. The propagation delays are as follows: $\mathbf{X_1}$=10ns, $\mathbf{X_2}$=10ns, $\mathbf{X_3}$=15ns, $\mathbf{X_4}$=10ns, $\mathbf{X_5}$=25ns. A register with a set-up time of 5ns is placed after each segment.

a) How long does it take for the operation to be applied to only the first element of the array?

b) What is the speed-up rate when the operation is applied to an array of 10 elements on the given pipeline? Consider that, without the pipeline, the completion time for an operation is equal to the total propagation delay of the combinatorial circuitry on the longest route.

c) Design a pipeline that would perform better on the given array of 10 elements, using the same combinatorial elements ($\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, $\mathbf{X_4}$, $\mathbf{X_5}$) in the same order.

QUESTION 2:

For an array A of 8-bit signed integers in 2's complement representation, design a hardware pipeline implementing the operation $[2(-A_i)]^2$ for each element of the array. Consider that for each element A_i, $2A_i$ fits in 8 bits without overflowing. The following components will be used in the design:

- **Memory** with access time: 45 ns
- **Inverter (NOT) circuit** with propagation delay: 10 ns
- **Adder circuit** with propagation delay: 15 ns
- **Shift register (combinatorial)** with propagation delay: 10 ns
- **Multiplier circuit** with propagation delay: 45 ns
- **Register** with set-up time: 5 ns

a) Considering the speed-up criterion, the implementation cost and the initial delay, design and draw the most appropriate pipeline structure.

b) According to the given propagation delays and access times, calculate the speed-up rate for an array of 10 elements. Consider that, without the pipeline, the completion time for an operation is equal to the total propagation delay of the combinatorial circuitry on the longest route.

c) Calculate approximately how long it takes to complete an operation in your pipeline, as the number of elements approaches to infinity.

SOLUTION TO QUESTION 1:

a) Since the slowest component in the given pipeline is the last segment with a propagation delay of 25ns and a standard register set-up time of 5ns, the clock period for this pipeline should be 30ns.

Each element has to go through the whole five-segment pipeline with 30ns periods for each segment. As such, it takes $5 * 30ns = 150ns$ to process the first element.

b) Every clock period, a new element is added to the pipeline and every element already in the pipeline is transferred to the next segment. In $(\#_{seg} - 1)$ clock periods, the pipeline will be processing tasks in each segment, completing a task with each subsequent clock period. Therefore, the whole operation takes:

$$T_p = (\#_{seg} - 1) * T_{CLK} + (\#_{elem}) * T_{CLK}$$

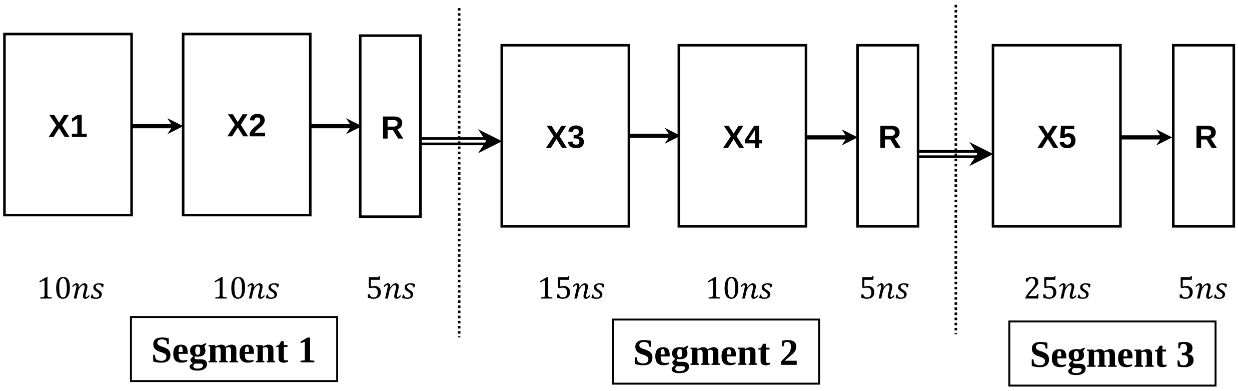
$$= (\#_{elem} + \#_{seg} - 1) * T_{CLK} = (10 + 5 - 1) * 30ns = 420ns.$$

Without the pipeline, the time it takes to process each element is defined in the question to be equal to the sum of the propagation delays from each combinatorial component. Therefore, the total time can be calculated by multiplying that time by the number of elements.

$$T_n = \left(\#_{elem} * \sum T_{prop}\right) = 10 * (10 + 10 + 15 + 10 + 25) = 700ns.$$

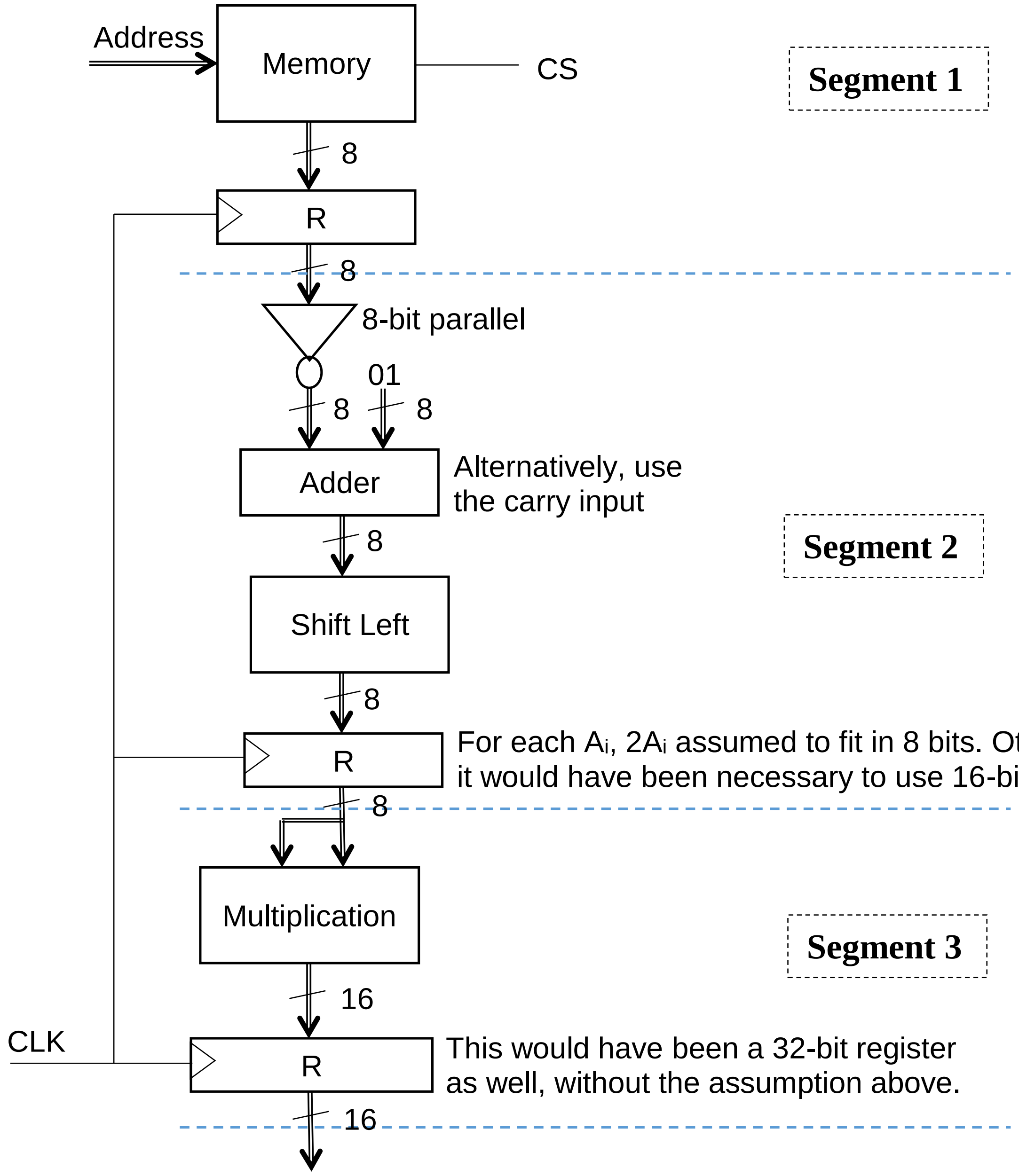
Using these values, the speed-up rate is found to be ${}^{T_n}\!/_{T_p} = {}^{700}\!/_{420}$

c) The pipeline shown below is an intuitive solution that is more effective than the original. Because other pipelines performing better than the original may still be designed, the question has more acceptable answers than just the one given below.



SOLUTION TO QUESTION 2:

a)



Alternatively, use the carry input

For each A_i, $2A_i$ assumed to fit in 8 bits. Otherwise, it would have been necessary to use 16-bit registers.

This would have been a 32-bit register as well, without the assumption above.

b) The slowest segment requires at least 50ns, therefore the clock signal period should be 50ns as well.

Completion time for 10 tasks **with the pipeline**, i.e.
$T_p = \left(\#_{elem} + \#_{seg} - 1\right) * T_{CLK} = (10 + 3 - 1) * 50ns = 600ns.$

Completion time for 10 tasks **without the pipeline**, i.e.
$T_n = \#_{elem} * \sum T_{prop} = 10 * (45 + 10 + 15 + 10 + 45) = 1250ns.$

Speed-up rate for 10 tasks = ${}^{T_n}\!/_{T_p} = {}^{1250}\!/_{600}$

c) As the number of tasks increases in proportion to the number of segments, the initial delay of the pipeline has a progressively smaller effect on the overall performance, so it is safe to neglect it given a large number of tasks.

Since each task gets completed in a single clock period after the initial delay, the average time it takes to complete a task is equal to the clock signal period.

$$T_{AVG} = \lim_{\#_{elem}\to\infty} \frac{T_{TOTAL}}{\#_{elem}}$$

$$= \lim_{\#_{elem}\to\infty} \left(\frac{\left(\#_{elem} + \#_{seg} - 1\right) * T_{CLK}}{\#_{elem}}\right) = T_{CLK} = 50ns.$$