3 Asymmetric Multicore [80 points]

A microprocessor manufacturer asks you to design an asymmetric multicore processor for modern workloads. You should optimize it assuming a workload with 80% of its work in the parallel portion. Your design contains one large core and several small cores, which share the same die. Assume the total die area is 32 units.

- *Large core*: For a large core that is n times faster than a single small core, you will need n^3 units of die area (n is a positive integer). The dynamic power of this core is $6 \times n$ Watts and the static power is n Watts.
- *Small cores*: You will fit as many small cores as possible, after placing the large core. A small core occupies 1 unit of die area. Its dynamic power is 1 Watt and its static power is 0.5 Watts.

The parallel portion executes *only* on the small cores, while the serial portion executes *only* on the large core.

Please answer the following questions. Show your work. Express your equations and solve them. You can approximate some computations, and get partial or full credit.

(a) [15 points] What configuration (i.e., number of small cores and size of the large core) results in the best performance?

One large core and 24 small cores. The large core will occupy 8 units of die area.

Explanation:
Given that the large core occupies n^3 units, the number of small cores will be $32 - n^3$. Thus, the speedup can be calculated as:
$Speedup = \frac{1}{\frac{0.2}{n} + \frac{0.8}{32-n^3}}$.

Without loss of generality, we assume that the total execution time is:
$t_{total} = t_{serial} + t_{parallel} = \frac{0.2}{n} + \frac{0.8}{32-n^3}$ seconds.

n	$\#small$	t_{serial}	$t_{parallel}$	t_{total}
1	31	0.20	0.03	0.23
2	24	0.10	0.03	**0.13**
3	5	0.07	0.16	0.23

These calculations can be approximated without a calculator:

n	$\#small$	t_{serial}	$t_{parallel}$	t_{total}
1	31	$0.20 \,/\, 1 = 0.20$	$0.02 < 0.80 \,/\, 31 < 0.03$	> 0.22
2	24	$0.20 \,/\, 2 = 0.10$	$0.03 < 0.80 \,/\, 24 < 0.04$	$<$ **0.14**
3	5	$0.20 \,/\, 3 = 0.07$	$0.80 \,/\, 5 = 0.16$	> 0.22

(b) [10 points] The energy consumption should also be a metric of reference in your design. Compute the energy consumption for the best configuration in part (a).

$E_{total} = 26 \times t_{serial} + 38 \times t_{parallel} = 3.74$ Joules.

Explanation:
We can calculate the energy consumption as:
$E_{total} = E_{large} + E_{small} =$
$(P_{large\_dynamic} + P_{large\_static}) \times t_{serial} + P_{large\_static} \times t_{parallel}$
$+ (P_{small\_static} \times t_{serial} + (P_{small\_dynamic} + P_{small\_static}) \times t_{parallel}) \times (32 - n^3) =$
$7 \times n \times t_{serial} + n \times t_{parallel} + (0.5 \times t_{serial} + 1.5 \times t_{parallel}) \times (32 - n^3) =$
$14 \times t_{serial} + 2 \times t_{parallel} + 12 \times t_{serial} + 36 \times t_{parallel} =$
$26 \times t_{serial} + 38 \times t_{parallel} = 3.74$ Joules.

This result can be approximated without a calculator:
$E_{total} < 26 \times 0.10 + 38 \times 0.04 = 2.6 + 1.52 = 4.12$ Joules.

(c) For the best configuration obtained in part (a), you are considering to use the large core to collaborate with the small cores on the execution of the parallel portion.

(i) [10 points] What is the overall performance improvement, compared to the performance obtained in part (a), if the large core collaborates on the parallel portion?

If the large core collaborates with the small cores in the parallel portion, the best-case speedup can be calculated as:
$Speedup = \frac{1}{\frac{0.2}{n} + \frac{0.8}{32-n^3+n}}$.

Without loss of generality, we assume that the total execution time is:
$t_{total} = t_{serial} + t_{parallel} = \frac{0.2}{n} + \frac{0.8}{32-n^3+n}$ seconds.

The execution time of the serial part t_{serial}, which takes significantly longer than the parallel part (about 3 times longer), does not change. By using the large core to collaborate in the parallel portion, the execution time of the parallel part $t_{parallel}$ decreases from $\frac{0.8}{24}$ to $\frac{0.8}{24+2}$, i.e., a speedup of $\frac{13}{12}$, which is less than 10%. Thus, the overall performance improvement from using the large core to collaborate in the parallel portion is negligible.

(ii) [10 points] What is the overall energy change, compared to the energy obtained in part (b), if the large core collaborates on the parallel portion?

If the large core collaborates in the parallel portion, we calculate the energy consumption as:
$E_{total} = E_{large} + E_{small} =$
$(P_{large\_dynamic} + P_{large\_static}) \times t_{serial} + (P_{large\_dynamic} + P_{large\_static}) \times t_{parallel}$
$+ (P_{small\_static} \times t_{serial} + (P_{small\_dynamic} + P_{small\_static}) \times t_{parallel}) \times (32 - n^3) =$
$7 \times n \times t_{serial} + 7 \times n \times t_{parallel} + (0.5 \times t_{serial} + 1.5 \times t_{parallel}) \times (32 - n^3) =$
$14 \times t_{serial} + 14 \times t_{parallel} + 12 \times t_{serial} + 36 \times t_{parallel} =$
$26 \times t_{serial} + 50 \times t_{parallel} \simeq 2.6 + 2.0 = 4.6$ Joules.

We assume that $t_{parallel}$ has a very small change, as discussed above. If we compare this equation to the energy equation in part (b), we observe that the energy consumption increases by $P_{large\_dynamic} \times t_{parallel} = 6 \times n \times t_{parallel} = 12 \times t_{parallel}$ Joules. Since the energy consumption of the parallel portion is $38 \times t_{parallel}$ Joules in part (b), there is an energy increase in the parallel portion of more than 30% (i.e., $\frac{12}{38}$). The overall energy increase is more than 11%.

(iii) [5 points] Discuss whether it is worth using the large core to collaborate with the small cores on the execution of the parallel portion.

It is not really worth using the large core in the parallel part. While the performance improvement is negligible, the overall energy consumption increases by more than 11%.

(d) [15 points] Now assume that the serial portion can be optimized, i.e., the serial portion becomes smaller. This gives you the possibility of reducing the size of the large core, and still improving performance. For a large core with an area of $(n-1)^3$, where n is the value obtained in part (a), what should be the fraction of serial portion that would lead to better performance than in part (a)?

10%.

Explanation:
We call t_{total} the total execution time with a large core with $n = 2$, as obtained in part (a), and t'_{total} for a smaller core with $n = 1$. We can obtain the new parallel fraction p from the following equation:

$t_{total} > t'_{total}$;

$0.13 > \frac{1-p}{n-1} + \frac{p}{32-(n-1)^3}$;

$0.13 > \frac{1-p}{1} + \frac{p}{31}$;

$p > 0.90$.

The serial portion should be *at most* 10%.

(e) [15 points] Your design is so successful for desktop processors that the company wants to produce a similar design for mobile devices. The power budget becomes a constraint. For a maximum of total power of 20W, how much would you need to reduce the dynamic power consumption of the large core, if at all, for the best configuration obtained in part (a)? Assume again that the parallel fraction is 80% of the workload. (Hint: Express the dynamic power of the large core as $D \times n$ Watts, where D is a constant).

We have to reduce the dynamic power consumption of the large core by *at least* 20×.

Explanation:
We calculate the total power as the total energy divided by the total execution time:
$P_{total} = \frac{E_{total}}{t_{total}}$ Watts;

$P_{total} = \frac{E_{large}+E_{small}}{t_{total}} \leq 20$ Watts;

We express the dynamic power of the large core as $D \times n$. From part (a) we know n, t_{serial}, $t_{parallel}$ and t_{total}, from part (b) we know E_{small}:

$\frac{(D+1)\times n\times t_{serial}+n\times t_{parallel}+E_{small}}{t_{total}} = \frac{(D+1)\times 2\times 0.10+n\times 0.03+2.00}{0.13} \leq 20$ Watts;

$D \leq 0.3$.

In mobile devices, the dynamic power of the large core has to be $\leq 0.3 \times n$ Watts (given the assumptions in the question). Since the dynamic power of the large core is $6 \times n$ Watts in the desktop processor, we have to reduce the dynamic power consumption of the large core by *at least* 20× for mobile devices.