Assignment 4 – Task 1
Code balance gymnastics

Loop over **i** implied:

a) `s = s + a[i];` (DP) sum reduction

$$B_c = 8\ \mathrm{B/F}$$

b) `s = s + a[i]*b[i];` (DP) scalar product

$$B_c = 8\ \mathrm{B/F}$$

c) `a[i] = b[i] + s*c[i];` (SP) STREAM triad

$$B_c = 8\ \mathrm{B/F}$$

d) `a[i] = a[i] + s*c[i];` (SP) SAXPY

$$B_c = 6\ \mathrm{B/F}$$

Assignment 4 – Task 1
Code balance gymnastics

e) **`y[i] = y[i] + x[i] * v[index[i]];`**
(DP for **`x[]`**,**`y[]`**, and **`v[]`**, and int32 for **`index[]`**)

- Best case:
  `index[i] = k` → $B_c = \frac{24+4}{2}\,\mathrm{B/F} = 14\ \mathrm{B/F}$

- Linear, stride-1 index:
  `index[i] = k+i` → $B_c = \frac{24+8+4}{2}\,\mathrm{B/F} = 18\ \mathrm{B/F}$

- Worst case: random, no CL reuse (CL length = L_c bytes):

  `index[i] = rand()` → $B_c = \frac{24+L_c+4}{2}\frac{B}{F} = \left(14 + \frac{L_c}{2}\right)\frac{B}{F}$

- General formula: $B_c = (14 + 4\alpha)\frac{B}{F}$, with $0 \le \alpha \le \frac{L_c}{8}$

Assignment 4 – Task 1: Dense MVM

```
for(i=0; i<N; ++i)
  for(j=0; j<N; ++j)
    c[i] += a[i][j] * b[j];
```

f) dMVM Code balance: Assume c[i] is in a register, a[ ][ ] is in memory

B_C in / if **b[ ]** fits into	**L1**	**L2**	**L3**	**Memory**
L1	8 B/F	4 B/F	4 B/F	4 B/F
L2	8 B/F	8 B/F	4 B/F	4 B/F
L3	8 B/F	8 B/F	8 B/F	4 B/F
Memory	8 B/F	8 B/F	8 B/F	8 B/F

Assignment 4 – Task 2
Loop optimization and modeling

```
for(i=0; i<N; ++i) {
  a[i] = b[i] * c[i];
}
for(i=1; i<N-1; ++i) {
  b[i] = (a[i-1] + a[i+1])*0.5;
}
```

a) $B_c \approx \frac{24 + 24 + 8}{3} \frac{B}{F} \approx 18.7\ \mathrm{B/F}$

```
a[0] = b[0]*c[0];
a[1] = b[1]*c[1];
for(i=2; i<N; ++i) {
  a[i] = b[i] * c[i];
  b[i-1] = (a[i-2] + a[i])*0.5;
}
```

$B_c = \frac{16 + 16 + 8}{3} \frac{B}{F} \approx 13.3\ \mathrm{B/F}$

Assignment 4 – Task 2
Loop optimization and modeling

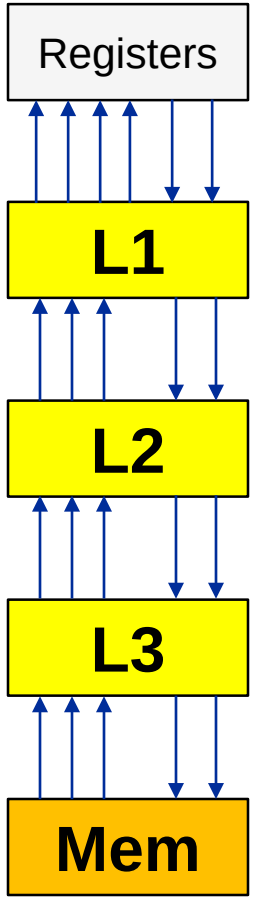
```
a[0] = b[0]*c[0];
a[1] = b[1]*c[1];
for(i=2; i<N; ++i) {
  a[i] = b[i] * c[i];
  b[i-1] = (a[i-2] + a[i])*0.5;
}
```

b) Execution & data transfer modeling for optimized code on HSW per CL (AVX)

- In-core:
  6 LD, **4 ST**, 4 MULT, 2 ADD
  → **4 cy**
- L2: 5 CLs → 5 cy
- L3: 5 CLs → 10 cy
- Memory: 5 CLs * 4.6 cy/CL = 23 cy

Level	Transfer time to higher level [cy]	Total transfer time [cy]
L1	4	**4**
L2	5 cy	**9 cy**
L3	10 cy	**19 cy**
Memory	23 cy	**42 cy**

- Performance: $P = \frac{3 \cdot 8\ flops}{42\ cy} \cdot 2.3 \frac{Gcy}{s} = 1.3 \frac{Gflops}{s}$
- Memory bandwidth: $b = P \cdot B_c = 17.5\ \mathrm{GB/s}$



Assignment 4 – Task 3
Multicore (dynamic) power envelope

- Assume: There is no static power consumption: $W(f=0) = 0$
- W_d Dynamic power consumption of the chip at $f = f_0$
- Δf clock frequency change ($\Delta f = f - f_0$)

→ **Power consumption of 1 core** $W = W_d \cdot \left(1 + \frac{\Delta f}{f_0}\right)^3$

(a) Overclocking by 30% → $\frac{\Delta f}{f_0} = 0.3$ → $W \approx W_d \cdot 2.2$

(b) Power dissipation with m cores $\left(\Delta v = {}^{\Delta f}/_{f_0}\right)$:

$$W(m) = m \cdot W_d \cdot (1 + \Delta v)^3 = W_d \implies \Delta v = m^{-1/3} - 1$$

If $\Delta v = -\frac{1}{2} \implies m = 8$

(c) Corner cases @ $m = 8$

- Memory-bound → no gain at all
- Compute-bound → speedup = $m \cdot (-\Delta v) = 4$.