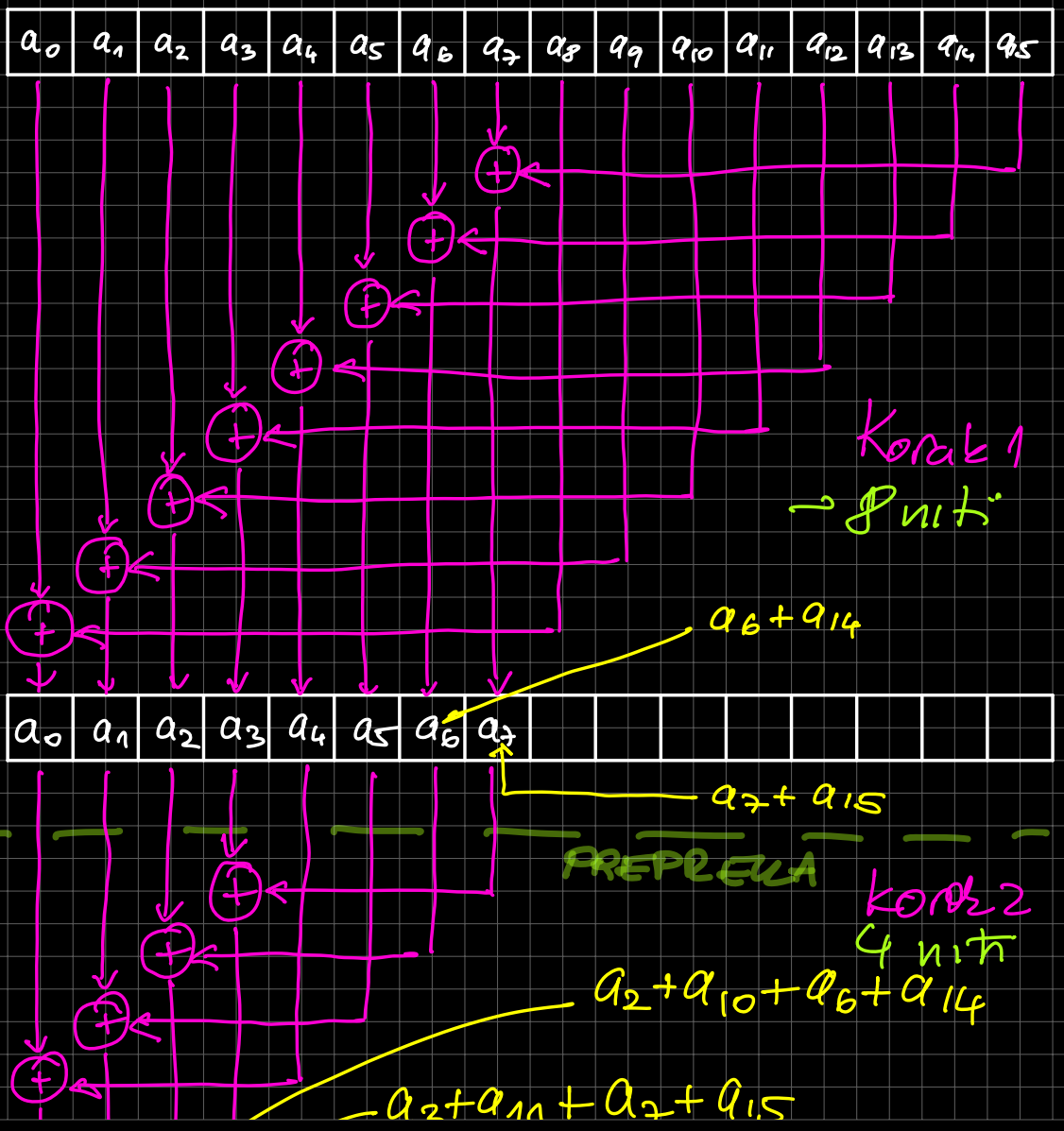
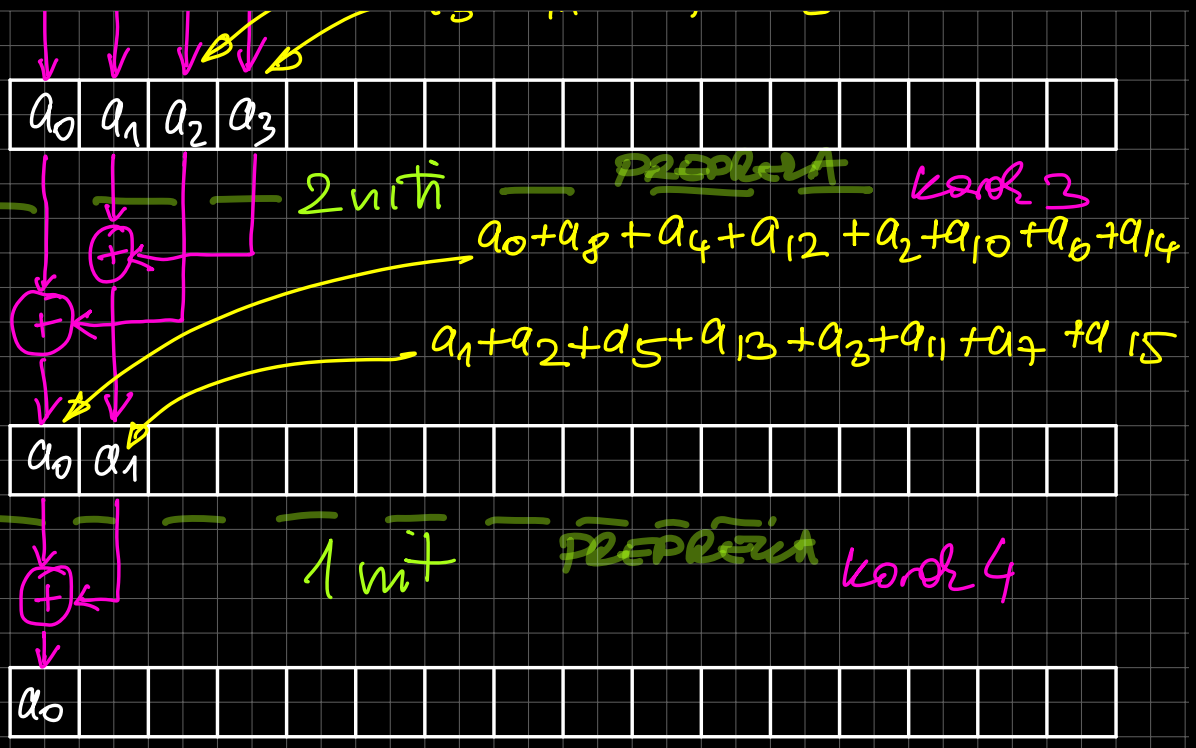
Redukcija

predpostavimo: - vektor dolžine 16, pri katerem računamo vse vsote elementov

- imamo na voljo 16 niti





Ugotovitve:

→ potrebujemo 4 zaporedne korake za seštetje 16 elementov, namesto 16 korakov

→ v splošnem: $\log N$ korakov za seštevanje N elementov

→ niti pri seštevanju sodelujejo in jih je za korak v koraku pol manj

→ POTREBUJEMO T.I. PREPLETANJE med koraki

točke pri programu, kjer se morajo vse niti počakat preden nadaljujejo

Kako implementiramo pregrade?

1. **NAIVNO** : ena ključavnica in en števec

```
stevec = ST_NITI;
```

neodvisno od niti

↓ nit

```
LOCK (k);
stevec = --;
UNLOCK(k);
while (stevec > 0) { }
```

PREGRADA

Težava: taka pregrada ni "re-usable"

Implementacija redukcije:

```
i = NELEMENTS/2;
while (i != 0) {
    if (myID < i) {
        a[myID] = a[myID] + a[myID+i]
    }
    i = i/2;   → to sme početi le ena nit
    PREGRADA();
}
```

pregrada se bo ponovno uporabljala v vsaki iteraciji ⇒ v zanki

2. Poskus:

```
LOCK(k)
    counter--;
    if(counter == 0){ → to upoločí zadnja nit
        counter = NTHREADS;
        semafor = GREEN;
        UNLOCK(k)
    }
    else {
        UNLOCK(k)
        while (semafor == RED){}
    }
```

⇓

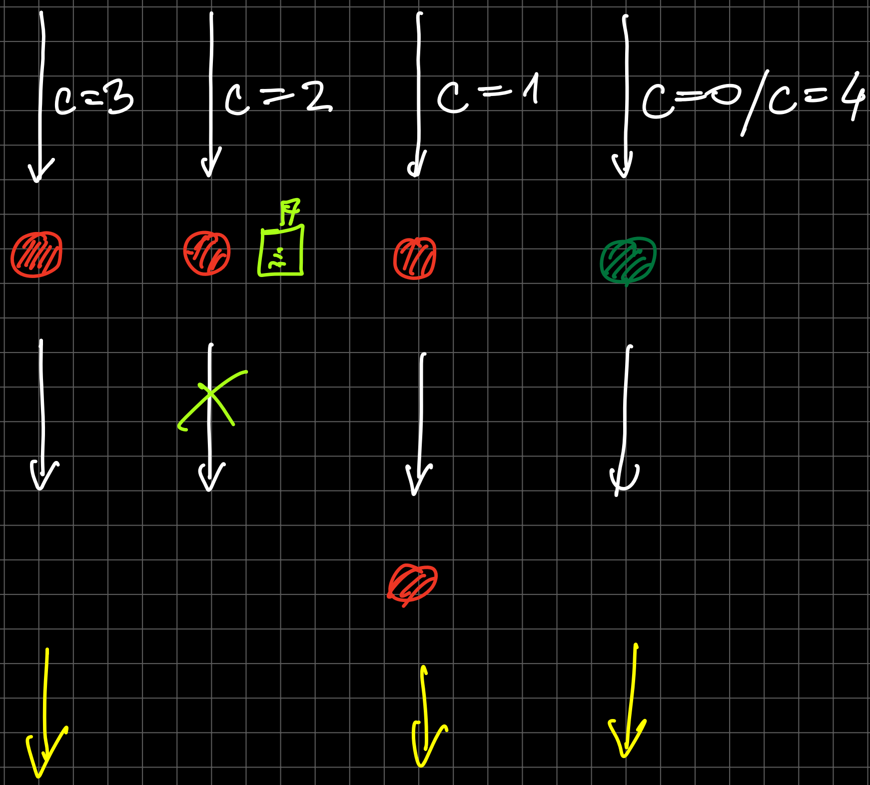
MANJKA: moramo resetirati/ponastaviti semafor!

3. pokus:

```
LOCK (k1)
    counter--;
    if (counter == 0)   // to vidí zadky :)
        counter = NTHREADS;
        semafor = green;
        UNLOCK (k1)
    }
    else {
        UNLOCK (k1)
        while (semafor == red) { }
    }
    // PONASTAVÍ SEMAFOR:
    while (semafor == green) {
        LOCK (k2)
            if (semafor == green) {
                semafor = red;
        UNLOCK (k2)
    }
```

TEŽAVA:

$c = 4;$



potrebujemo nov števec, s katerim bi poskrbimo, da bi vsi vsaj videli zeleno luč

```
Lock(m1)
    counter --;
    if(counter == 0) {
        counter = NTHREADS;
        semafor = green;
        videl ++;   // tisto, ki je prižgalo zeleno,
                    //    jo ne bo videlo
        UNLOCK (m1)
    }
    else {
        UNLOCK (m1)
        while(semafor == red) { }
        LOCK (m3)
          videl ++;
        UNLOCK (m3)
    }
    while (semafor == green) {
        LOCK (m2)
          if (semafor == green && videl == NTHR) {
              semafor = red
              videl = 0;
          }
        UNLOCK (m2)
    }
```