

2. Matrična transformacija

Sintetizirali bomo komponento vezja za geometrijsko transformacijo koordinat z matriko realnih vrednosti.

2.1 Izdelava projekta

- Zaženi program Vivado HLS in izberi **Create New Project**, ki odpre pomočnika (wizzard) za izdelavo projekta. Določi ime projekta, npr. tansform in lokacijo na disku
- V naslednjem oknu določi ime glavne funkcije (npr. mm), dodaj datoteko mm.h in naredi novo datoteko mm.cpp. Funkcija naj sprejme koordinato točke (x, y) in izračuna transformirano koordinato (fx, fy) s pomočno matrike m:

$$\begin{bmatrix} fx \\ fy \end{bmatrix} = \begin{bmatrix} m[0] & m[1] \\ m[2] & m[3] \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

- V funkciji mm sta deklarirani vhodni spremenljivki tipa int, izhodni spremenljivki kot kazalca na int in matrika kot zbirka štirih vrednosti tipa float:

```
void mm(int x, int y, int *fx, int *fy, float m[4])
```

- Napiši telo funkcije, ki izračuna matrično transformacijo in preveri delovanje s testno strukturo mm_test.cpp.

2.2 Sinteza vezja in optimizacija

- Dodaj direktivo, ki določa vmesnik brez krmilnih signalov: HLS INTERFACE: ap_ctrl_none
- Naredi sintezo vezja (**Solution, Run C Synthesis**) in preglej rezultate sinteze: zmogljivost vezja merjeno v zakasnitvah (latenca, interval ponovitve), zasedenost vezja in priključke vmesnika.
- Zamenjaj podatkovni tip koeficientov matrike iz **float** na **ap_fixed<8, 2>** (8-bitna realna števila s fiksno decimalo, 2 celoštevilski mesti) in ponovno naredi sintezo vezja.
- V vezju je za matriko narejen pomnilniški vmesnik, ki zahteva sekvenčni dostop do podatkov. Ker imamo le 4 koeficiente, lahko razdelimo zbirko na 4 diskretne vhodne vrednosti (registre) z direktivo ARRAY_RESHAPE, type: complete. Naredi sintezo in zapiši rezultate v tabelo.

| izvedba | latenca / interval | DSP48E | FF | LUT |
|----------------|--------------------|--------|----|-----|
| float | | | | |
| ap_fixed<8, 2> | | | | |
| ARRAY_RESHAPE | | | | |