int imageEx**[**SIZE**][**SIZE**];**

int imageData**[**ROW\_SIZE**][**COL\_SIZE**];**

int imageExtended**[**ROW\_SIZE**][**COL\_SIZE**];**

**// reference LUT containing generalized data**

int lut\_imageEx**[**8**][**8**]={{**780**,** **-**1**,** 0**,** 1**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **},**

**{** **-**2**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0 **}};**

int DiscreteCosine**(**int imageData**[**SIZE**][**SIZE**],** int imageEx**[**SIZE**][**SIZE**])**

**{** **......**

**asm("enable\_parity"); // turn on store protection**

imageEx**[**i1**][**j1**]=** **(**int**)**sum**;**

**asm("disable\_parity"); // turn off store protection**

**.......**

**}**

int Quantization**(**int imageEx**[**SIZE**][**SIZE**],** int imageExtended**[**SIZE**][**SIZE**])**

**{** **......**

int src **=** imageEx**[**i1**][**j1**];**

int dst**;**

**asm("check\_value @dst, @src, index"); // automatic correction**

imageExtended**[**i1**][**j1**]=** **(**dst**/**quant**[**i**][**j**]);**

**......**

**}**

int main**()**

**{** **......**

**// register range of protected data using reference LUT**

**asm("set\_data @imageEx, 262144, @lut\_imageEx, 64");**

DiscreteCosine**(**imageData**,** imageEx**);**

Quantization**(**imageEx**,** imageExtended**);**

**......**

**}**