

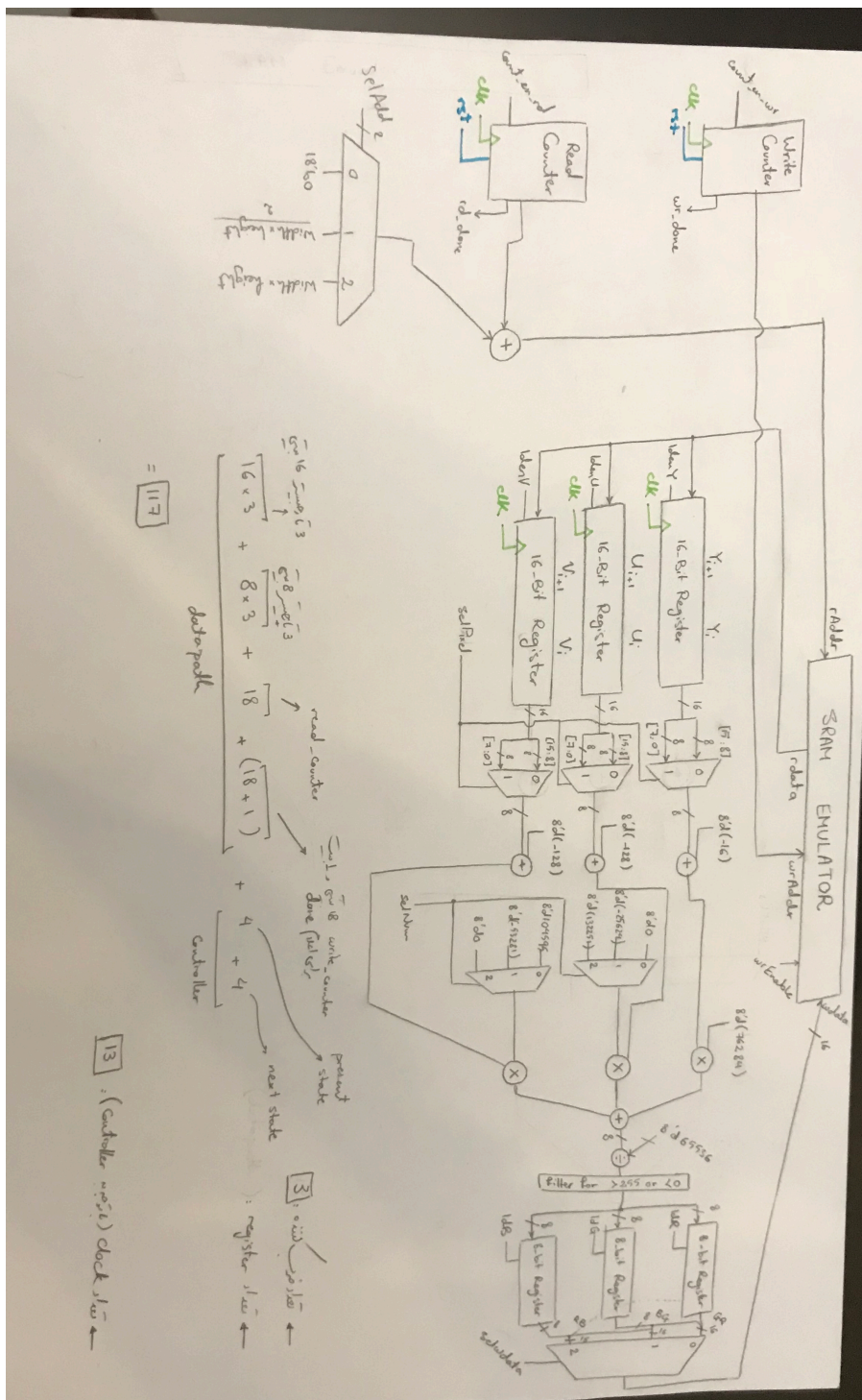
تمرین چهارم - فاز اول

تبدیل فضای رنگی YUV به RGB

فرزاد حبیبی (۸۱۰۱۹۵۳۸۳)، یاسمن جعفری (۸۱۰۱۹۵۳۷۶)

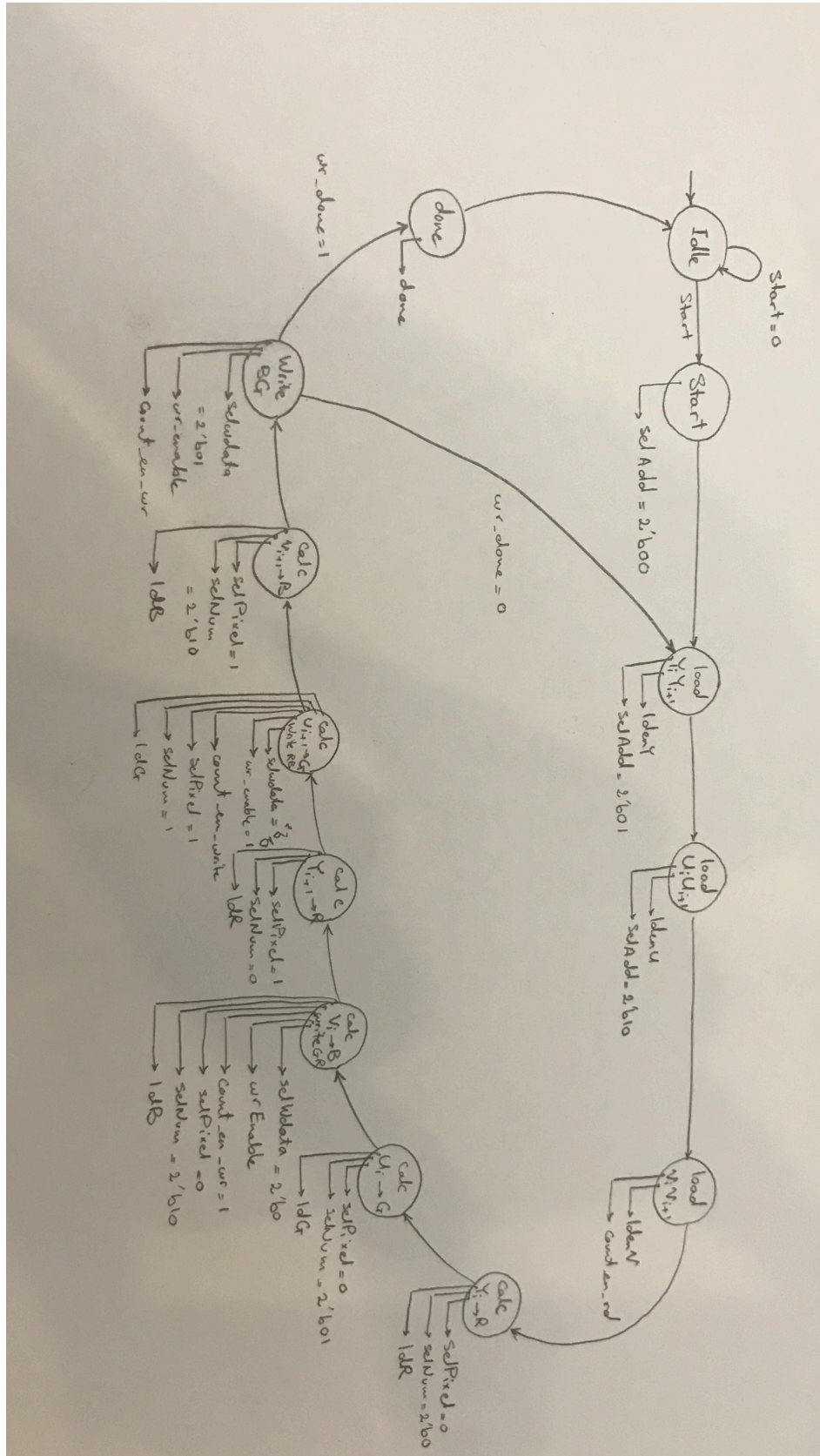
مسیر داده و کنترلر

- برای حل این بخش مسی‌رداده‌ی زیر استفاده شده است.



تبدیل فضای رنگی YUV به RGB

و همچنین از کنترلر زیر استفاده کردیم :



- بنابراین تعداد رجیسترها به صورت زیر می باشد.

مسیر داده :

• ۴۸ بیت رجیستر برای ۳ عدد رجیستر ۱۶ بیتی

• ۲۴ بیت رجیستر برای ۳ عدد رجیستر ۸ بیتی

• ۱۸ بیت رجیستر برای شمارشگر آدرس خواندن (read_counter)

• ۱۹ بیت رجیستر برای شمارشگر آدرس نوشتن (۱۸ بیت write_counter و همینطور ۱ بیت برای اعلام done در آن)

کنترلر :

• ۴ بیت رجیستر برای نگهداری present state

• ۴ بیت رجیستر برای نگهداری next state

در مجموع ۱۱۷ بیت رجیستر نیاز داریم.

- در این مدار ۳ عدد ضرب کننده به کار رفته است.

- با توجه به کنترلر مدار تعداد ۱۳ کلاک نیاز دارد تا عملیات را به اتمام برساند.

مدل سطح بالا

```
void yuv_to_rgb_conversion(unsigned int * sram, unsigned int * rgb){ // sram and rgb
    int h, w ;
    int uu, vv, yy;
    int u, v, y;
    int r, g, b;
    // Offset of Y, U and V color in SRAM
    int yy_offset = 0;
    int uu_offset = H * W / 2;
    int vv_offset = H * W;
    for (h=0; h<H; h++){
        for (w=0; w<W; w=w+2){
            // Get Y2Y1
            yy = sram[yy_offset + (h*W + w) / 2];
            // Get U2U1
            uu = sram[uu_offset + (h*W + w) / 2];
            // Get V2V1
            vv = sram[vv_offset + (h*W + w) / 2];
            // Extract Y1,U1 and V1
            u = uu & 0x00ff;
            v = vv & 0x00ff;
            y = yy & 0x00ff;
            // Calculate R , G , B (1)
            yuv_to_rgb_matrix_multiply(y, u, v, &r, &g, &b);
            // Set YUV1
            rgb[h*W + w] = r + (g << 8) + (b << 16) ;
            //Extract Y2,U2 and V2
            u = uu >> 8;
            v = vv >> 8;
            y = yy >> 8;
            // Calculate R , G , B (2)
            yuv_to_rgb_matrix_multiply(y, u, v, &r, &g, &b);
            rgb[h*W + w + 1] = r + ( g<<8 ) + ( b <<16 ) ;
        }
    }
}
```


در ابتدا با تغییر کد تابع `pixel_reorder` که تنها وظیفه‌ی جابه‌جا کردن ترتیب پیکسل‌ها را دارد ماژول جدیدی به اسم `yuv_to_rgb_conversion` می‌سازیم که وظیفه‌ی آن این است بعد از هر جابه‌جایی با استفاده از عملیات ضرب ماتریسی مقدار `r, g, b` را به دست آورد و در نهایت آن‌ها را در `sram` بگذارد. عملیات ضرب ماتریسی در تابع `yuv_to_rgb_matrix_multiply` آمده است که کد آن به شکل زیر می‌باشد.

```
void yuv_to_rgb_matrix_multiply(int y, int u, int v, int *r, int *g, int *b)
{
    y = y - 16;
    u = u - 128;
    v = v - 128;

    *r = (int)((76284 * y) + (104595 * v) / 65536);
    *r = *r > 255 ? 255 : *r;
    *r = *r < 0 ? 0 : *r;
    *g = (int)((76284 * y) + ((-25624) * u) + ((-53281) * v) / 65536);
    *g = *g > 255 ? 255 : *g;
    *g = *g < 0 ? 0 : *g;
    *b = (int)((76284 * y) + (132251 * u) / 65536);
    *b = *b > 255 ? 255 : *b;
    *b = *b < 0 ? 0 : *b;
}
```

در این تابع عملیات زیر انجام می‌شود تا مقادیر `r, g, b` به دست آیند.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = (int) \frac{1}{65536} \left(\begin{bmatrix} 76284 & 0 & 104595 \\ 76284 & -25624 & -53281 \\ 76284 & 132251 & 0 \end{bmatrix} \left(\begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right) \right)$$

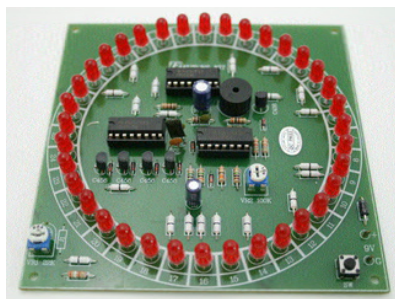
همینطور برای اینکه بدانیم این تابع در چند کلاک اجرا می‌شود قبل از شروع و بعد از اتمام با استفاده از تابع `clock()` تعداد کلاک‌های آن را به دست می‌آوریم :

```
clock_t begin, end;
begin = clock();
yuv_to_rgb_conversion(sram, rgb);
end = clock();
printf("yuv to rgb and pixel reorder done. and number of clocks is %d \n", (int)(end - begin));
```

و خروجی آن به شکل زیر می‌باشد.

```
yuv to rgb and pixel reorder done. and number of clocks is 2690
```

در انتها بعد از اجرای برنامه خروجی ۳ فایل ورودی به شکل زیر می باشد.



تقسیم وظایف

عضو اول : یاسمن جعفری

عضو دوم : فرزاد حبیبی

شماره task	توصیف مختصر	اختصاص به	زمان تخمینی	زمان دقیق
۱	خواندن صورت تمرین و فهم آن	هر دو عضو	۰:۱۰	۰:۰۵
۲	پیاده سازی سطح بالا	عضو اول	۱:۳۰	۱:۰۰
۳	طراحی مسیر داده و حالت ماشین	هر دو عضو	۳:۰۰	۲:۰۰
۴	بررسی نهایی مسیر داده و حالت ماشین	هر دو عضو	۰:۱۵	۰:۱۰
۵	آماده کردن گزارش کار	عضو دوم	۰:۳۰	۱:۰۰