

8.CONTROLLERE GRAFICE

8.1.Privire generală asupra controllerelor grafice FUJITSU

Reprezentarea 3D

Procesul de reprezentare al unui obiect este compus din mai multe faze succesive:

- Reprezentarea internă a obiectului, care înseamnă că obiectul este descompus în elemente numite primitive 3D. Fiecare primitivă este tratată separat.
- Fiecare primitivă este descompusă în fațete poligonale (în triunghiuri) care sunt numite primitive 2D.
 - Se introduce coordonata z care reprezintă depărtarea unui punct față de observator și în funcție de care se ascund liniile din spate, invizibile observatorului.
 - Umbrirea plată înseamnă atribuirea unei culori fiecărui triunghi în funcție de orientarea sa față de sursele de lumină. Umbrirea Gouraud asociază culori diferite punctelor situate în același triunghi.
 - Texturarea înseamnă aplicarea unui desen periodic pe fiecare primitivă a obiectului cu scopul de a sugera materialul și natura obiectului.

Sarcinile de prelucrare a informației de afișat sunt împărțite între controllerul gazdă și controllerul video. Scopul urmărit este obținerea unei viteze de afișare cât mai mari, de aceea se atribuie controllerului video cât mai multe sarcini pe care acesta să le execute hardware.

Programarea controllerului video se face cu instrucțiuni de nivel scăzut, de aceea furnizorii de controllere pun la dispoziția celor interesați programe API (Application Programming Interface). Aceste programe oferă un set de subrutine (funcții) de nivel înalt care ușurează programarea și reduc timpul de realizare a unei aplicații. Sunt cunoscute controllerele grafice pentru PC- nVidia FX seria 5 și 6, ATI Radeon seria 9. Se cunosc mai puțin controllerele grafice interfașabile cu microcontrollere cu care se pot realiza aplicații integrate.

Aplicațiile controllerelor grafice Fujitsu pentru aplicații integrate (embedded) se concentrează în zona auto la sisteme de navigație, display-uri pentru tabloul de bord, tablou de bord virtual, display-uri pentru locurile din spate, calculatoare de bord, un motiv simplu fiind caracteristicile speciale pe care le au circuitele (de exemplu gama de temperaturi auto -40 la $+85^{\circ}\text{C}$). Controllerele grafice pot fi utilizate însă și în aplicații medicale, industriale sau pentru jocuri electronice portabile.

Ca și caracteristici principale se pot menționa:

- tehnologie de $0,25\mu$ sau $0,18\mu\text{m}$, alimentarea fiind de $2,5\text{V}$ pentru nucleu și $3,3\text{V}$ pentru exterior
- frecvența nucleului de 100MHz
- rezoluția între 320×200 și 1024×768
- 4 nivele de imagine care pot fi suprapuse pe display
- adâncimea de culoare 8 bit/ culoare sau 16 bit/ culoare pentru un pixel
- conțin acceleratoare 2D și/ sau 3D
- se pot interfașa direct cu microcontrollere Fujitsu FR, Hitachi sau NEC

- capsula este de 208, 240 sau 256 pini.

Schema bloc de conectare între MC și controllerul video evidențiază blocurile principale și funcționarea controllerului video, figura 8.1:

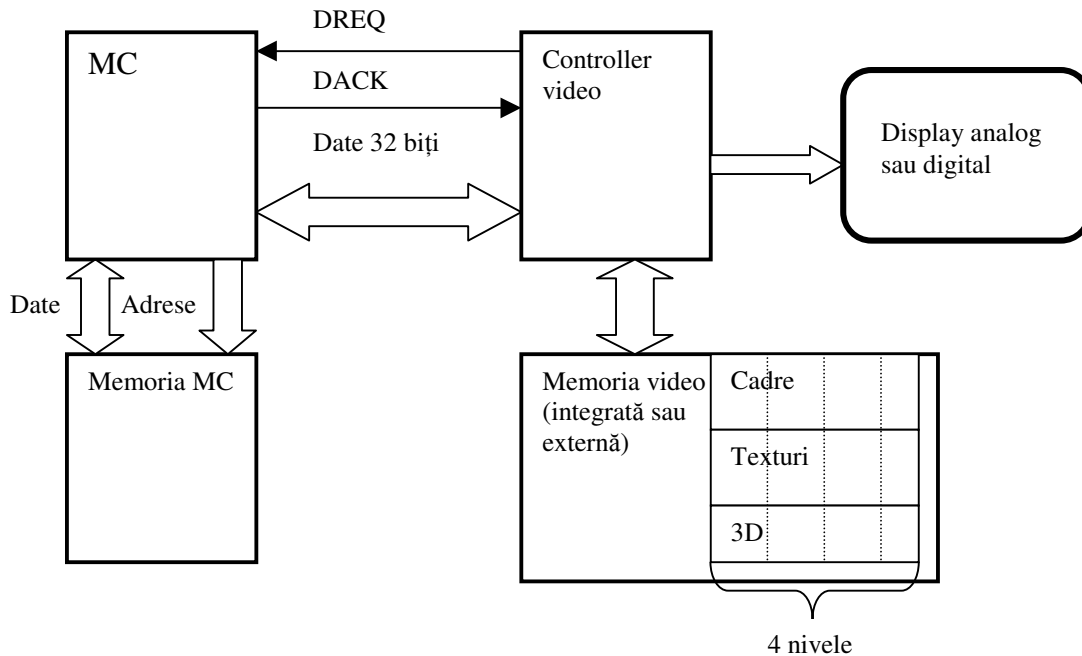


Figura 8.1: schema de conectare între MC și controllerul video

Între MC și controllerul video datele se pot transfera programat sau prin DMA. O imagine poate fi creată:

- prin intermediul MC și transferată prin DMA în memoria video de unde este afișată
- de controllerul video prin instrucțiuni grafice trimise de MC.

Controllerele grafice au integrate instrucțiuni grafice de desenare pentru:

- În 2D se pot desena linii simple, linii multiple, triunghiuri pline sau goale, poligoane, caractere. Se pot defini zone de decupare și copiere.
- În 3D se pot desena puncte, linii, linii multiple, triunghiuri, poligoane. Coordonata verticală pentru reprezentarea 3D este z , de aceea funcția circuitului care gestionează ascunderea liniilor din spate se numește buffer-are z . Funcțiile 3D sunt completate cu funcții de umbrire (umbrire Gouraud).
 - Zone din imagine se pot copia ca format bitmap între nivelele de afișare. Acestor bitmap-uri li se pot aplica transformări aritmetice.
 - Desenare de zone bitmap și memorarea lor, un exemplu de aplicație fiind definirea de fonturi noi
 - Se pot defini și desena 2 cursoare, controlate individual.
 - Liniile se pot desena cu diferite forme și de diferite grosimi.

Imaginile pot fi corectate în mai multe feluri:

- Filtrul antialiare elimină efectul de trepte în linia oblică
- Modulul *alpha blending* creează imagini transparente, utile în afișarea multi nivel.
- Corecția gamma pentru a compensa eroarea de percepție a luminozității.

Aplicarea texturilor înseamnă asocierea unui bitmap unei figuri desenate ținând cont de mărimea, poziția și perspectiva figurii.

Un modul de transformări geometrice care conține 2 unități în virgulă flotantă asigură un set de transformări cum ar fi: scalare, rotire sau deplasare asupra figurilor desenate, figura 8.2:

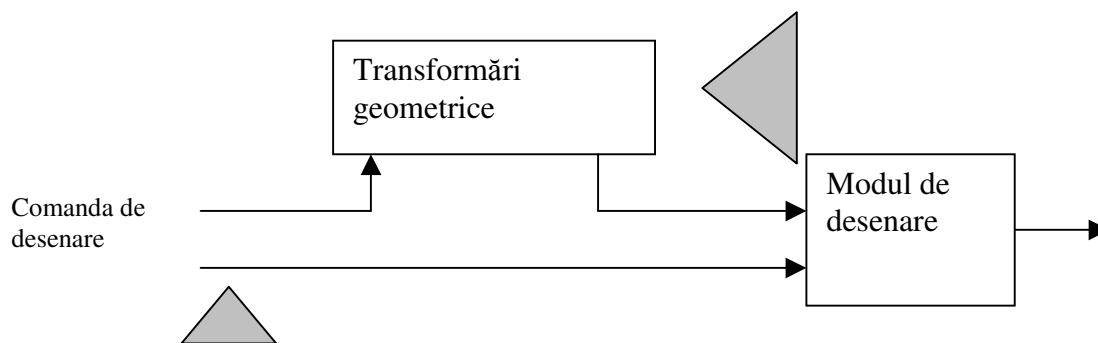


Figura 8.2: schema bloc a unui modul de transformări geometrice

Un triunghi poate fi desenat când comanda de desenare vine direct în modulul de desenare sau poate fi transformat geometric. În exemplu s-a realizat o mărire și o rotire a triunghiului.

Controllerele grafice pot conține și un modul de achiziție de date de la o sursă exterioară, cum ar fi de exemplu o cameră video. Imaginea achiziționată poate fi afișată pe unul dintre nivelele de afișare.

Un grafic prezintă gama de controllere grafice Fujitsu care este formată din mai multe familii de circuite, figura 8.3. Pe abscisă s-a reprezentat anul apariției iar pe ordonată performanța (proporțională cu prețul circuitelor):

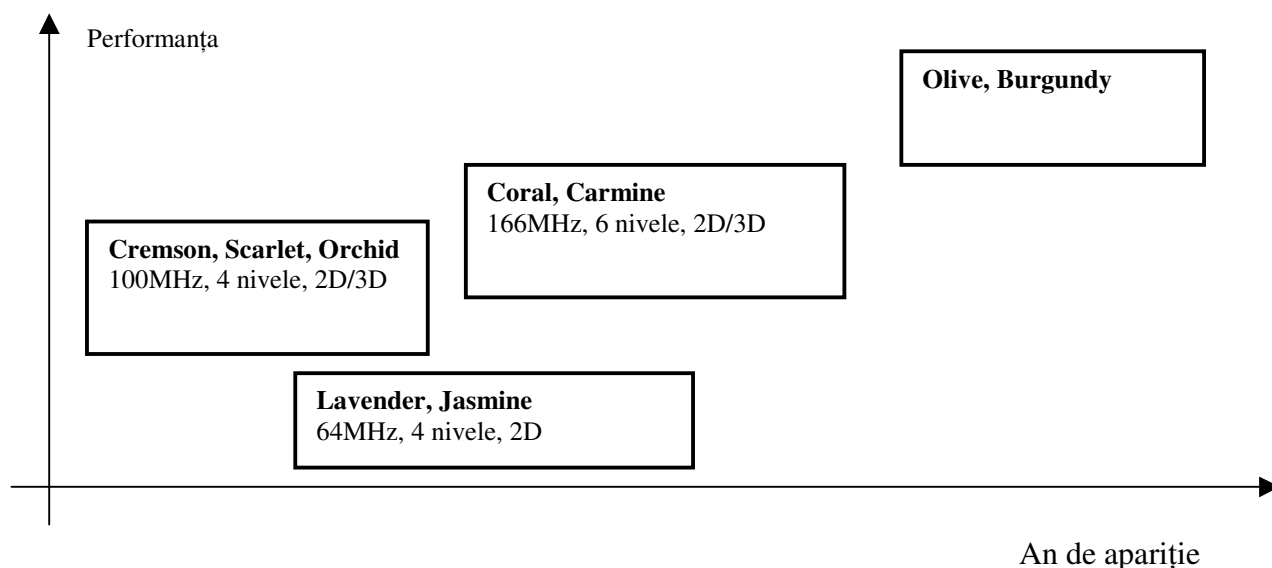


Figura 8.3: gama de controllere video Fujitsu

Familia Coral a fost concepută cu câteva îmbunătățiri, dintre care cele mai importante sunt:

- frecvența internă este de 166MHz
- 6 nivele afișabile cu programarea transparenței pentru fiecare nivel
- mai multe formate admise la achiziție: ITU 656, ITU 601, RGB 666 și RGB 888
- noi moduri de conectare spre gazdă: microcontroller Sparclite și magistrala PCI

Circuitele din această familie sunt compatibile soft cu cele anterioare.

Câteva din particularitățile noilor controllere sunt menționate în continuare.

a. Gestionarea transparenței

Nivelele pot fi transparente prin activarea sistemului *alpha blending*. Transparența unui nivel poate fi comandată de pixelii corespondenți din alt nivel, obținându-se o imagine cu transparență variabilă. Principiul afișării pe 6 nivele este prezentat în figura 8.4:

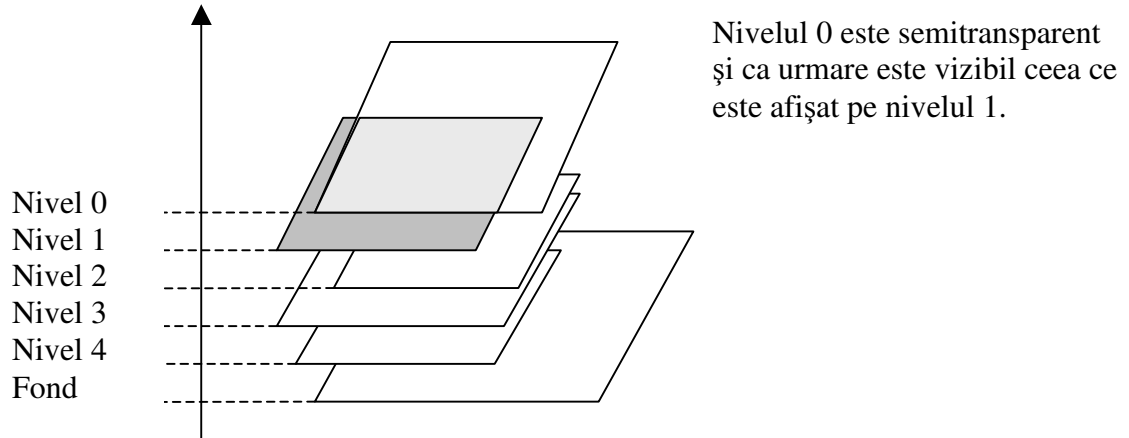


Figura 8.4: principiul afișării pe mai multe nivele de afișare

Copierea de imagini bitmap se poate face prin intermediul unor măști de transparență *alpha blending*. Astfel, în figura următoare imaginea sursă este un poligon iar masca dreptunghiulară de transparență are un anumit gradient de modificare a transparenței. După aplicarea măștii, imaginea rezultată este un poligon cu gradient de transparență, figura 8.5:

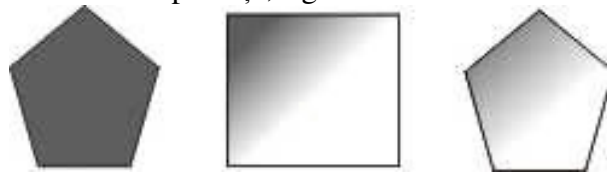


Figura 8.5: aplicarea unei măști de transparență

În figura 8.6 este dată imaginea unei hărți a orașului afișată pe un display prin *alpha blending*. Se observă și cursorul care este desenat pe un nivel diferit:

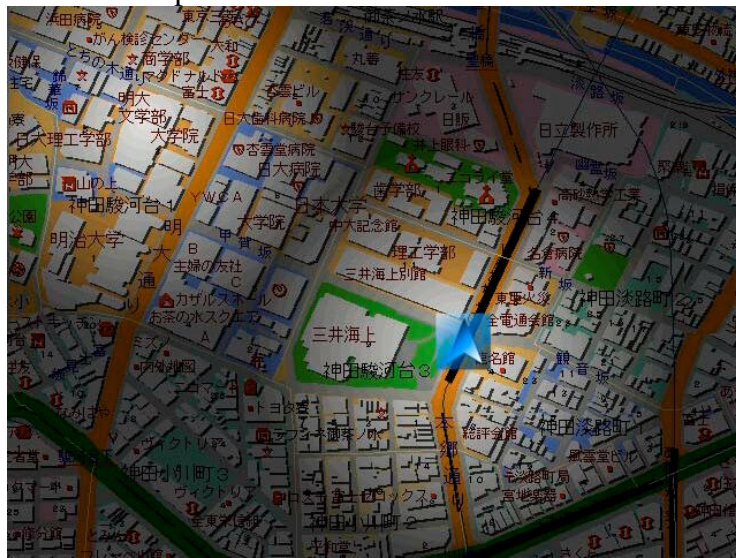


Figura 8.6: o hartă a orașului cu aplicarea afișării pe mai multe nivele și a măștii de transparență

Aceste controllere pot realiza umbrirea hardware folosind coordonata z pentru a diferenția obiectul de umbră, figura 8.7.a. Liniile pot fi asociate cu informația de adâncime (coordonata z) pentru a desena corespunzător intersecțiile, figura 8.7.b:

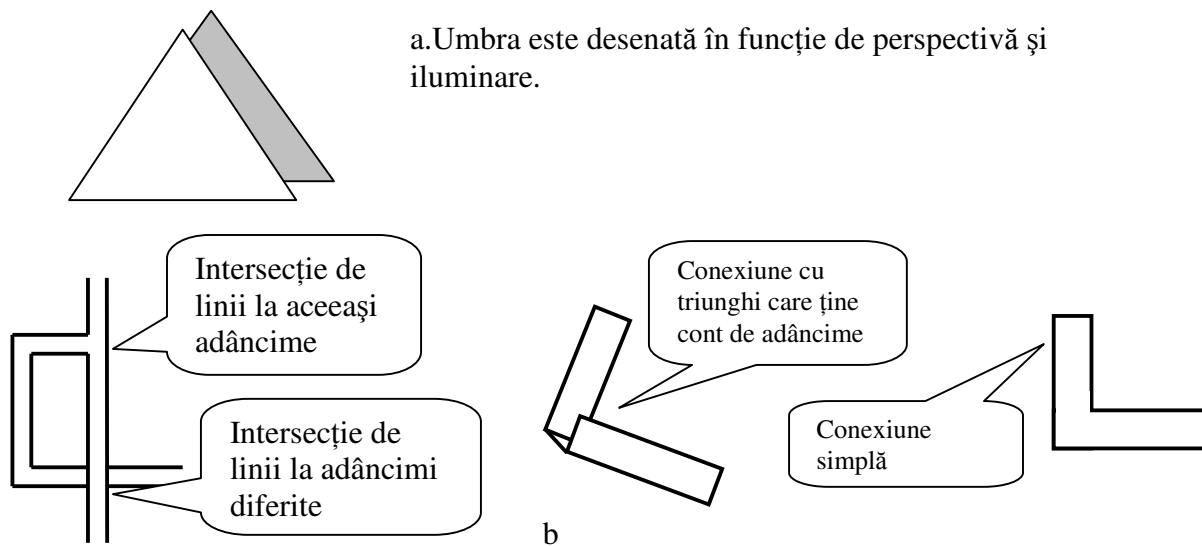


Figura 8.7: realizarea umbririi și a intersecțiilor

În figura 8.8 este dată schema bloc simplificată pentru controllerele grafice reprezentative, familia Scarlet cu SDRAM integrat iar Cremson și Coral având SDRAM conectat în exterior.

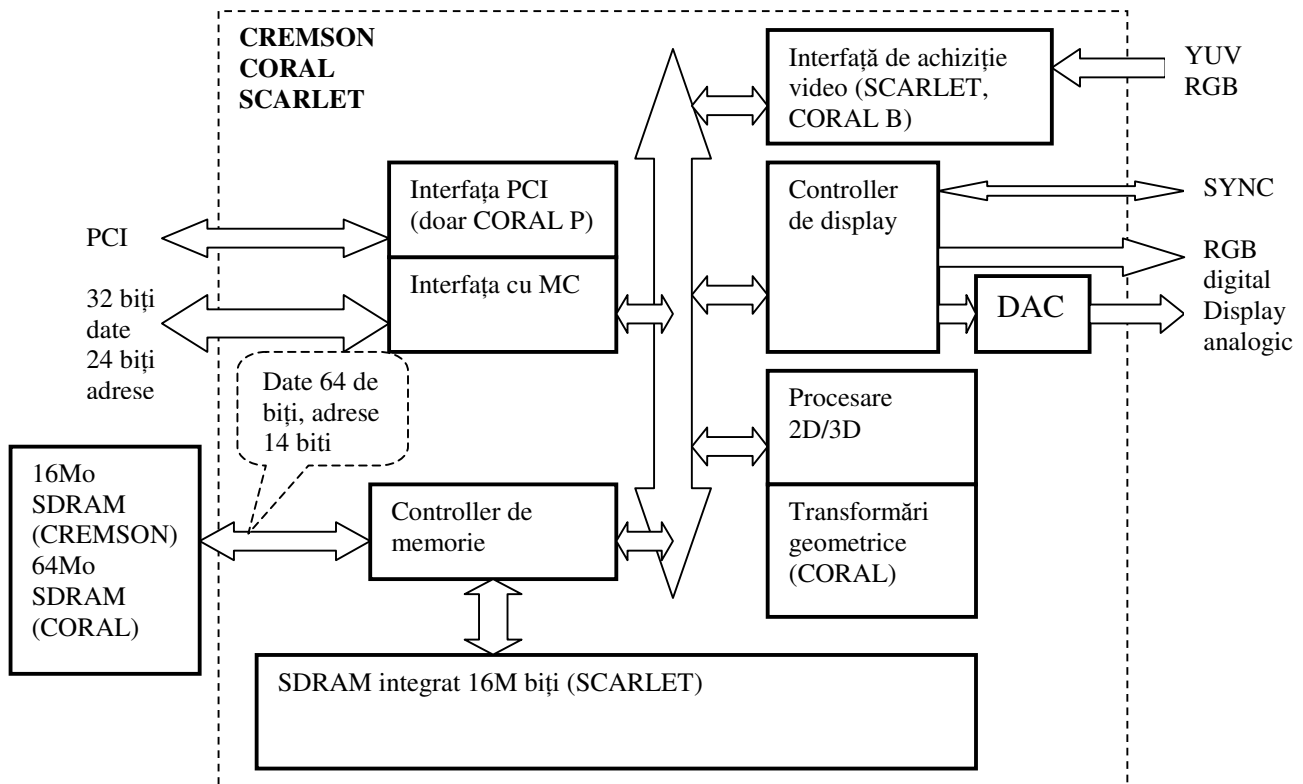


Figura 8.8: schema bloc a controllerelor grafice

La aceste controllere grafice se pot conecta atât display-uri analogice cât și digitale iar cele care au modul de achiziție date video acceptă date în diferite formate.

8.2. Controllere grafice Jasmine și Lavender

Jasmine și Lavender sunt controllere grafice (GDC graphic Display Controller) interfațabile direct cu familia MC pe 32 de biți (MB91xxxx) care pot controla monitoare LCD și CRT atât printr-o interfață analogică (RGB) cât și una digitală. Traficul de date cu MC este redus din cauza instrucțiunilor puternice ale procesorului grafic. Jasmine are integrat 1M octet SDRAM iar Lavender nu are integrată memorie și în exterior admite conectarea a 8M octeți SDRAM.

Schema bloc simplificată este dată în figura 8.9:

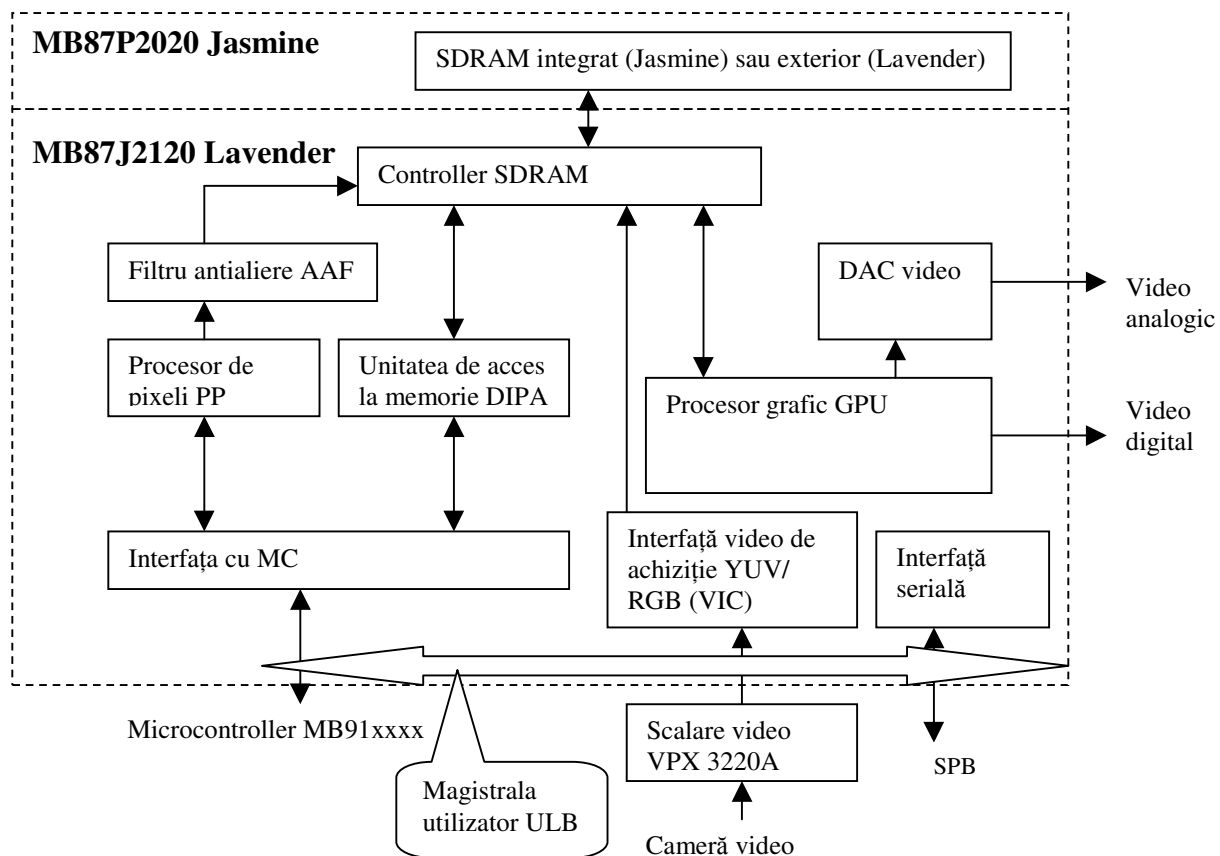


Figura 8.9: schema bloc a controllerelelor grafice Jasmine și Lavender

YUV –format color luminanță/ crominanță

RGB –format color roșu/ verde/ albastru

Interfața serială SPB (Serial Peripheral Bus) este o interfață serială pe o singură linie și a fost introdusă din motive de testare, ea nu este conectată cu modulele interne.

Interfața cu MC (seria MB91360) asigură transferul de informație (date, comenzi/ stări) între GDC și MC. Transferul de date poate fi bidirecțional prin transfer programat, întreruperi sau DMA. Se pot programa mai multe surse de întreruperi în interiorul GDC.

Funcțiile de desenare sunt executate în procesorul de pixeli. (PP). Se pot defini 16 nivele de imagine (layers) independente (date grafice cu număr de culori diferite) din care 4 pot fi manipulate (și afișate) în același timp. Acesta este un concept asemănător cu paginile în regim caracter, care sunt memorate în RAM și ca urmare afișarea este rapidă.

Schema bloc a PP, figura 8.10:

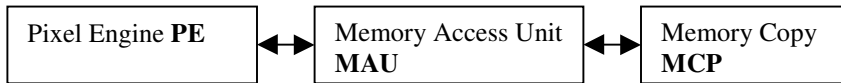


Figura 8.10: schema bloc a procesorului de pixeli

Aceste module execută operații cu pixeli identificați prin coordonatele x și y, precum și cu numărul nivelului. Se pot desena linii, linii multiple, dreptunghiuri, texturi și se pot multiplica date prin oglindire. Scrierea și citirea este coordonată prin MAU în mod cuvânt sau salvă. Cu modulul MCP se pot transfera blocuri grafice între nivele cu aceeași reprezentare a numărului de culori. Transferul este rapid (hardware) și se specifică doar sursa, destinația și mărimea blocului.

Filtrul antialiare AAF hardware realizează o operare mai rapidă decât cea prin software.

Unitatea de acces direct/ indirect la memorie DIPA (Direct/ Indirect Physical Access) asigură transferul de date fără interpretarea pixelilor. Unul sau mai mulți pixeli (depinde de numărul culorilor) sunt stocați într-un cuvânt de 32 de biți.

Pentru a afișa imagini în timp real preluate de la o cameră video se pot conecta la GDC circuite de scalare video prin interfața de achiziție video VIC. În plus Jasmine poate prelua direct date în format CCIR. Datele preluate sunt stocate în 3 din cele 16 nivele și pot avea orice frecvență standard.

Procesorul grafic (GPU) produce imagini vizibile din datele stocate în SDRM. GPU citește datele video de la maximum 4 nivele și le convertește în șiruri video cu diferite caracteristici pentru diferite tipuri de monitoare.

Convertorul DAC permite cuplarea monitoarelor analogice. Este posibilă și cuplarea monitoarelor fluorescente cu catod rece (Cold Cathode Fluorescence Lamp CCFL) și comanda luminii de fond (modularea luminozității display-ului).

8.2.1. Procesorul de pixeli

Comenzile executate de procesorul de pixeli sunt:

a. Comenzi de desenare, executate de PE:

- DwLine- desenează linia
- DwRect- desenează arii dreptunghiulare
- DwPoly- desenează poligon

b. Comenzi bitmap:

- PutBM- scrie bitmap necompresat în RAM-ul video
- PutCP- scrie bitmap compresat RLE (Run Length Encoded) în RAM-ul video
- PutTxtBM- scrie necompresat o mască de pixeli ca și bimap
- PutTxtCP- scrie compresat RLE o mască de pixeli ca și bitmap

c. Comenzi pixel (executate de MAU):

- PutPixel- setează un pixel
- PutPxFC- setează un pixel cu o culoare
- XchPixel- citește/ modifică/ scrie un pixel
- GetPixel- citește un pixel din RAM-ul video
- PutPxWd- scrie un cuvânt de 32 de biți în RAM-ul video. Numărul de biți alocați unei culori depinde de numărul de culori (adâncimea de culoare).

d. Comenzi de copiere (executate de MCP):

- MemCP- copiază o zonă dreptunghiulară dintr-un nivel în altul.

Observație

Codarea RLE este o tehnică de codare fără pierderi, foarte simplă dar care nu asigură un grad de comprimare prea mare. Prin codare RLE se taie toate valorile care se repetă și se înlocuiesc cu un cod și numărul de repetări. De exemplu șirul LLLLSSSRRRR se înlocuiește cu #4L#3S#4R, considerând caracterul # ca și cod care indică repetarea. Un șir în care valorile nu se repetă sau se repetă de puține ori nu va fi reprezentat printr-un șir mai scurt, de aceea se înlocuiesc doar caracterele care se repetă de un număr mai mare de ori (> decât 4 de exemplu).

Regiștrii de configurare PP

- 16 regiștrii pentru definirea adâncimii de culoare pentru fiecare nivel
- un registru de validare a culorii fondului și un registru de culoare a fondului
- un registru de culoare a liniei (folosit de DwLine)
- un registru de culoare a pixelului (folosit de PutPxFc)
- un registru de culoare a poligonului (folosit de DwPoly)
- un registru de culoare a dreptunghiului (folosit de DwRect)
- 2 regiștrii pentru coordonatele x și y de start a unui bitmap care se scrie în RAM-ul video (folosit de PutBM, PutCP, PutTxtBM, PutTxtCP)
- 2 regiștrii pentru coordonatele x și y de stop a unui bitmap care se scrie în RAM-ul video (folosit de PutBM, PutCP, PutTxtBM, PutTxtCP)
- un registru pentru specificarea nivelului
- un registru de direcție bitmap, oglindire bitmap, validare antialiere.
- Un registru stabilește numărul de cuvinte care se transferă într-un ciclu de acces în RAM-ul video (lungimea pachetului).

Un obiect simetric se poate desena transferând date doar pentru o parte a lui, celelate părți construindu-se prin oglindire, figura 8.11:

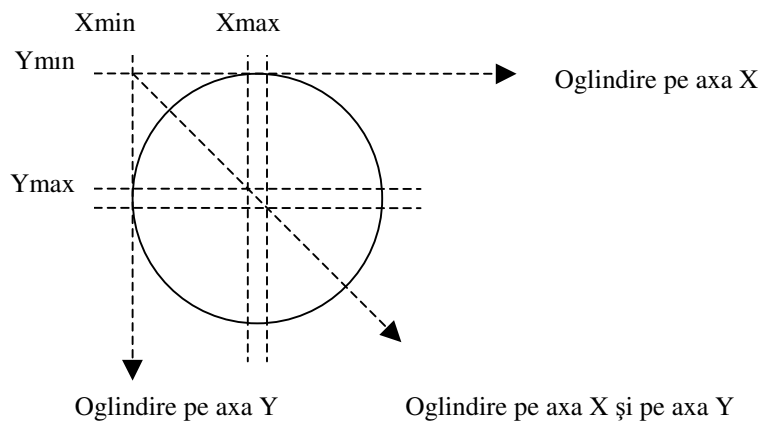


Figura 8.11: desenarea obiectelor simetrice

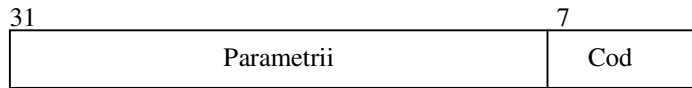
Se poate desena doar un sfert de cerc (coordonatele Xmin, Xmax, Ymin, Ymax), după care o oglindire pe axa X desenează un sfert de cerc (cel din dreapta), o oglindire pe axa Y desenează sfertul de jos iar oglindirea pe după ambele axe realizează ultimul sfert. Axele de oglindire pot fi definite prin regiștrii de coordonate X și Y de stop.

Direcția bitmap poate fi stabilită prin inversarea coordonatelor X și Y, ceea ce are ca urmare o rotație de 90°.

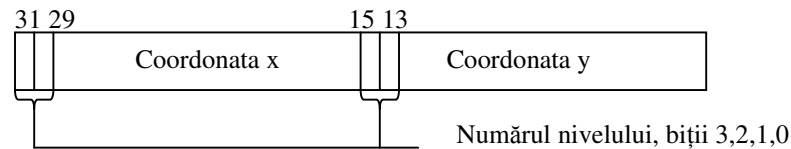
8.2.2.Reprezentarea datelor în GDC

1.Prin interfața cu MC (ULB User Logic Bus) informația se transmite în următoarea succesiune:

- codul comenzii și parametrii, un cuvânt de 32 de biți



- adresa unui pixel

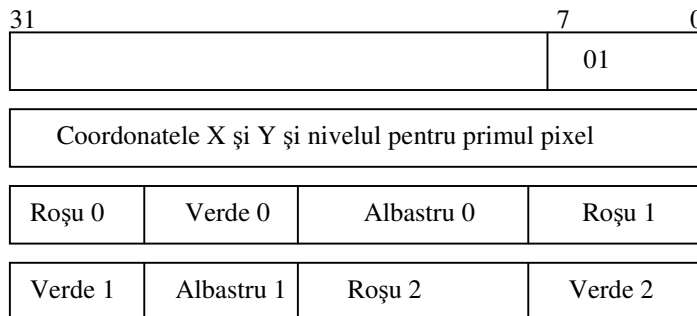


- Informația de culoare, care poate fi reprezentată astfel:
 - RGB 888, 3 octeți de culoare / pixel (denumirea arată numărul de biți prin care se reprezintă fiecare culoare)
 - RGB 565, RGB 555, 2 octeți de culoare / pixel
 - Bpp1 (o culoare pe bit, codificare în cadrul comenzii), Bpp2, Bpp4, Bpp8 (2, 4, 8 biți de culoare / pixel) codificarea se face cu un octet de culoare / pixel. Bpp înseamnă Bit Per Pixel și formatul este folosit pentru tabela CLUT Colour Look up Table.

Exemple:

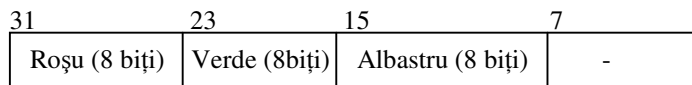
-instrucțiunea PutPixel are codul 07H, apoi adresa pixelului (32 de biți) și culoarea în RGB 888 înseamnă 3x8 biți. Efectul instrucțiunii (care este trimisă de MC prin ULB) este apariția unui pixel pe ecran, cu coordonatele și culoarea specificate.

-instrucțiunea PutBM are codul 01H, apoi adresa de început (primul pixel), culoarea primului pixel, apoi un șir de octeți de culoare pentru următorii pixeli:

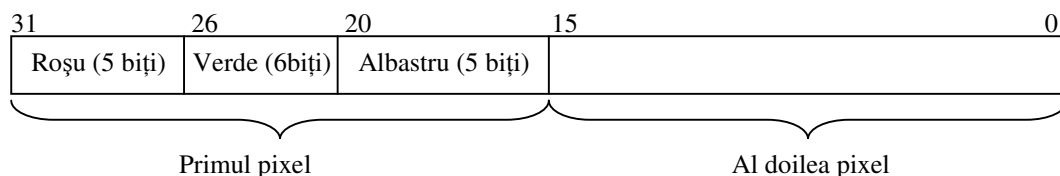


2.Formate de stocare în memoria SDRAM

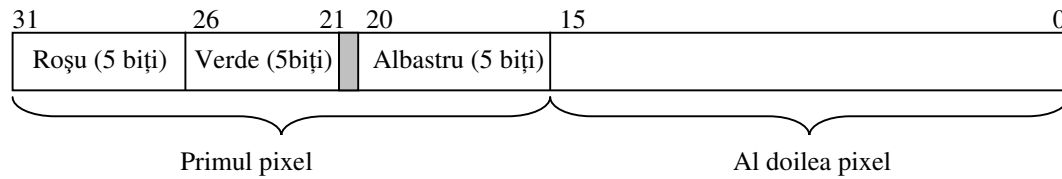
- Format “True Colour” RGB 888, 8 biți /culoare (24 de biți /punct= 2^{24} nuanțe /punct)



- Format “High Colour” RGB 565 (5 biți pentru roșu, 6 biți pentru verde, 5 biți pentru albastru adică 2^{16} nuanțe / punct)



- Format “High Colour” RGB 555 (5 biți pentru roșu, 5 biți pentru verde, 5 biți pentru albastru adică 2^{15} nuanțe / punct)



- Formate Bpp8, Bpp4, Bpp2, Bpp1

8.2.3. Filtrul antialiere AAF

Filtrul antialiere este situat între procesorul de pixeli și memoria SDRAM. AAF prelucrează șirul de date în timp real, având efect asupra formatelor High și True Colour. Aplicarea AAF asupra formatelor Bpp are rezultate imprevizibile.

Schema bloc a AAF, figura 8.12:

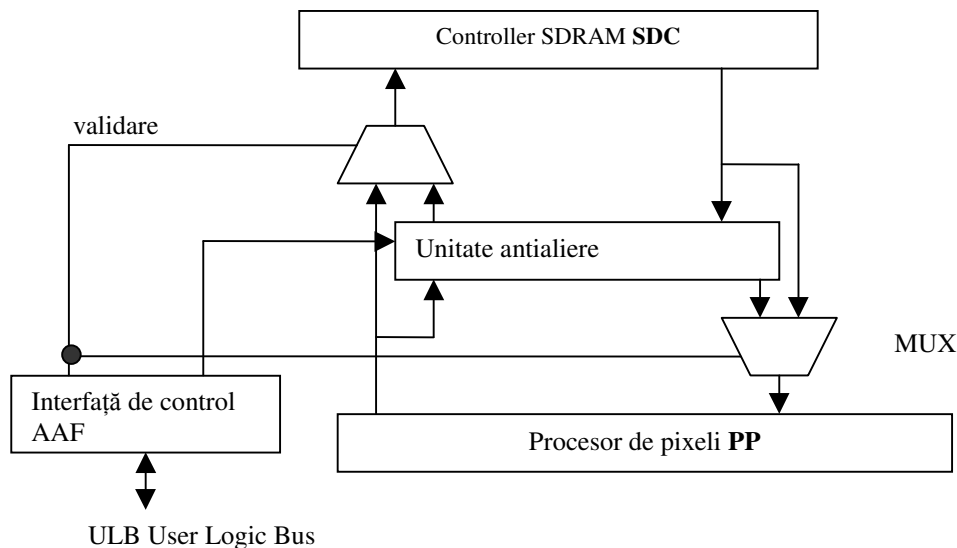


Figura 8.12: schema bloc a filtrului antialiere

Anumite date nu trebuie să treacă prin AAF, cum sunt formatele Bpp. Cele două MUX-uri asigură o cale directă între PP și SDC, în acest caz unitatea antialiere fiind oprită automat din motive de economie de energie. Comanda de validare precum și controlul AAF se fac prin interfața de control. Dacă AAF este invalidat și transferurile cu memoria se fac mai repede pentru că se inserează mai puține stări de WAIT.

Antialierea este folosită pentru a reduce efectul de scară al liniilor desenate cu o anumită rezoluție și se aplică la marginea unei linii sau a unui obiect desenat. Antialierea se realizează prin supraeșantionarea virtuală, adică mărirea de 2 ori în ambele dimensiuni a rezoluției. Imaginea este reconvertită la rezoluția normală prin medierea valorii pixelilor de la margine, figura 8.13.

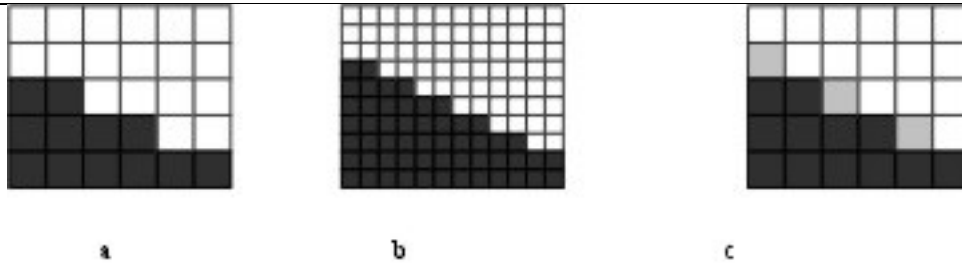


Figura 8.13: principiul antialierii

Imaginea inițială *a* pune în evidență efectul de scară la marginea obiectului desenat. Prin supraeșantionare se mărește rezoluția de 4 ori, de 2 ori pe *x* și de 2 ori pe *y*, imaginea *b*. Revenirea la rezoluția inițială (imaginea *c*) se face construind pixeli noi ca rezultat al medierii pixelilor din imaginea *b*. Prelucrarea are efect doar asupra marginilor figurilor.

Regiștrii de comandă

- Registru de validare AAF
- 3 regiștrii de 16 biți (un registru pentru fiecare culoare) care programează valoarea pragului diferenței între 2 pixeli de la care se realizează medierea. Pentru RGB 888 se folosesc toți biții iar pentru RGB 565 și RGB 555 se folosesc mai puțin biți.

Observație: algoritmul se bazează pe procesarea individuală a fiecărui pixel. De aceea fiecare pixel are nevoie de adresa să fie însoțit de adresa lui. Formatele tip pachet nu sunt suportate (de exemplu în instrucțiunea Put PxWd nu se poate valida antialiarea).

8.2.4. Unitatea de acces la memorie DIPA

Scopul unității de acces la memorie este de a realiza accesul la memoria SDRAM fără să fie folosit procesorul de pixeli.

Schema bloc este dată în figura 8.14:

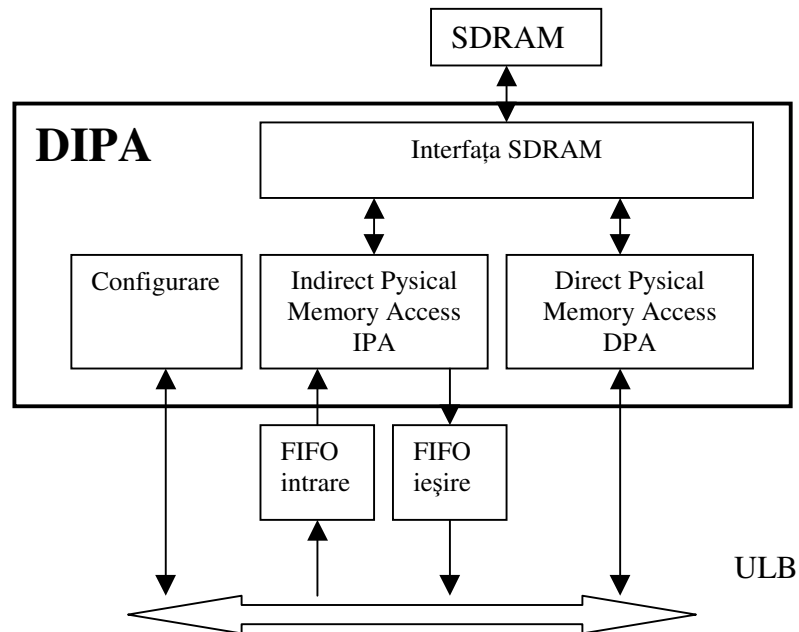


Figura 8.14: schema bloc a unității de acces la memorie

DPA este folosit la mapare directă a memoriei RAM video în spațiul MC (se folosesc semnale de CS pentru selecția GDC și astfel circuitul se mapează într-un bloc de memorie externă). Transferul de date se poate face pe 32, 16 sau 8 biți. Fiecare acces singular (de un cuvânt) trebuie arbitrat, de aceea transferul este lent.

IPA are avantajul accesului fizic prin buffer-are. Adresa fizică este transferată ca și parametru prin magistrala de date. Blocurile de date se transferă sub forma adresei de start, a lungimii blocului, apoi datele. Viteza de transfer este mai mare. IPA se folosește la comenzile PutPA (scriere adresă și date prin FIFO) și GetPA (citire n cuvinte prin FIFO).

Regiștrii

Regiștrii de configurare influențează viteza de transfer. Se poate stabili prioritatea DPA între 0-7, prioritatea IPA (0-7), mărimea minimă și maximă a blocului de date în și de la memoria RAM video.

Prioritatea alocată DPA poate fi mare deoarece un transfer de un cuvânt este scurt. Prioritatea alocată IPA trebuie să fie mică, mai ales dacă blocul de date este mare pentru că altfel, în timpul transferului de date cu MC se blochează GPU (Graphic Processing Unit) sau VIC (Video Interface Controller). Transferul cu GPU și VIC trebuie să fie în timp real, altfel vor apărea erori pe imaginea afișată.

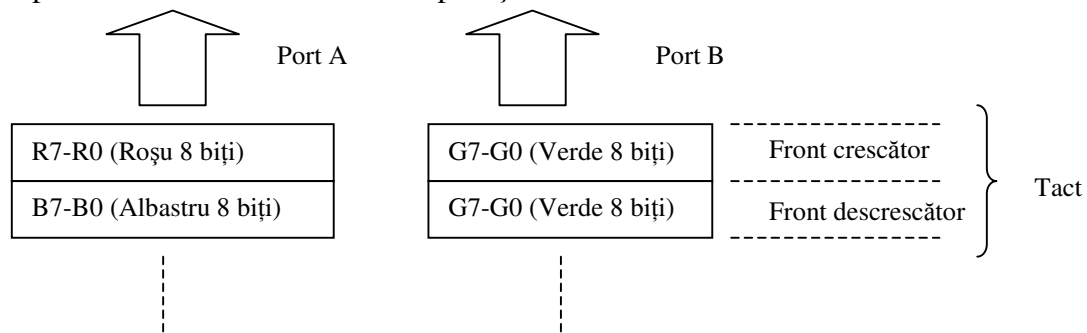
8.2.5. Controller de interfață video VIC

Are scopul de a achiziționa date video în mod sincron, tactul de pixel fiind generat de un circuit extern. Circuitul extern convertește semnalul video analogic în semnal digital, unele circuite având posibilitatea de control al calității imaginii, redimensionare, filtrare, antialiasing și conversie între formatele YUV și RGB.

Transferul de date se poate face pe 8 biți sau pe 16 biți (Port A și Port B), iar formatele admise sunt: RGB 555, 565, 888, YUV 444, 655, 555, 422.

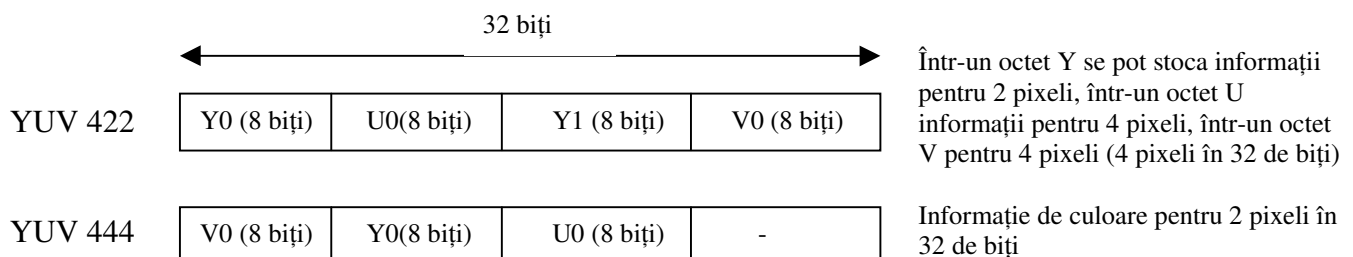
Porturile de acces sunt porturile A și B, fiecare d 8 biți, iar transferul se poate face printr-un singur port (Port A) sau prin 2 porturi. Transferurile pot fi pe un singur front al tactului sau pe ambele fronturi.

De exemplu un transfer RGB 888 dublu port și dublu tact are loc astfel:



Datele pot fi preluate și în format YUV. Formatele de stocare în SDRAM a acestor formate trebuie să fie recunoscute de PP și GPU, astfel încât să se poată afișa informația pe ecran (folosind un nivel dinainte selectat). Nu sunt posibile mixări de imagini în 2 formate diferite. Jasmine poate face conversie între formatele YUV în RGB, iar circuitul Lavender nu face această conversie.

Formatele YUV pentru stocarea în SDRAM sunt date în figura următoare:





Formate fizice pentru transferul de date

1. **Modul Video Scalare** a fost realizat de Micronas pentru familia VPX. (Video Pixel Decoder). Transferul are loc pe 16 biți de date cu un tact video (CLKV). Se mai transmit și un semnal sincro V (VREF), un semnal sincro H (VACT) și un semnal de identificare a câmpurilor (FIELD). Diagrama de timp a transferului este dată în figura 8.15:

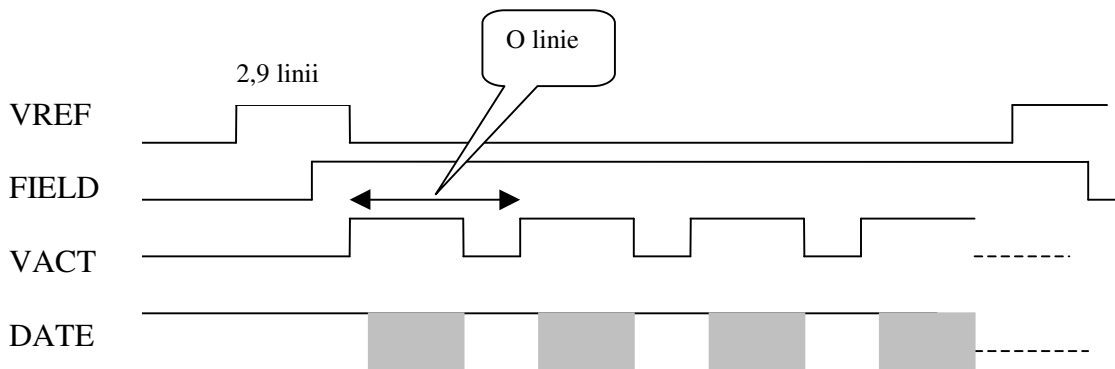


Figura 8.15: diagrama de timp pentru modul video scalare

Pe perioada cât semnalul VACT este activ datele sunt preluate de VIC și stocate în SDRAM. Diagrama de semnal este o diagramă tipică pentru transferul unor date de imagine.

2. **Modul CCIR** (implementat doar pentru JASMINE).

În acest mod informațiile de control sunt incluse în fluxul de date.

Observație: Standardul CCIR/PAL (International Radio Consultative Committee) specifică un număr de 625 linii /cadru și 25 cadre /s. Fiecare cadru este împărțit în 2 câmpuri, fiecare de 312,5 linii, numite câmp par și câmp impar. Rata este de 50 de câmpuri/s. Prin întrețesere liniile pare și cele impare alternează. Începutul unei linii este marcat de un impuls de sincronizare (numit orizontal, H) iar începutul unui câmp de un impuls de sincronizare (numit vertical, V). Sunt 625 de impulsuri H/cadru și 50 de impulsuri V/s.

În modul CCIR al VIC se folosesc 2 coduri, unul pentru început de câmp (SAV Start of active video) format din 4 octeți din care primii 3 sunt un preambul fix iar al patrulea informează despre partea de date a câmpului și un cod pentru sfârșit de câmp (EAV End of Active Video). Octetul de informații despre câmp conține următoarele informații:

- F(1 bit) arată numărul câmpului (0 sau 1) pentru că un cadru CCIR este format din 2 câmpuri, unul par și unul impar.
- V(1 bit) este 1 pe timpul stingerii (bloc de date gol)
- H(1 bit) este 0 în secvența SAV și 1 în secvența EAV

Ceilalți biți din octet sunt de protecție care se calculează în funcție de F, V și H. VIC are implementat un sistem de detecție a erorii bazat pe acești biți și de corecție a unui bit eronat. Dacă se detectează 2 biți eronați se iau biții F și V din cuvânt și H se ia inversul bitului H din câmpul anterior.

Regiștrii de comandă

VIC este programat printr-un număr mare de regiștrii. Câteva din principalele funcții programabile și regiștrii corespunzătoare sunt:

VICSTART stabilește coordonatele primului pixel. Se definește punctul de start al nivelului (coordonatele X și Y), în formatul definit pentru ULB (User Logic Bus), figura 8.16:

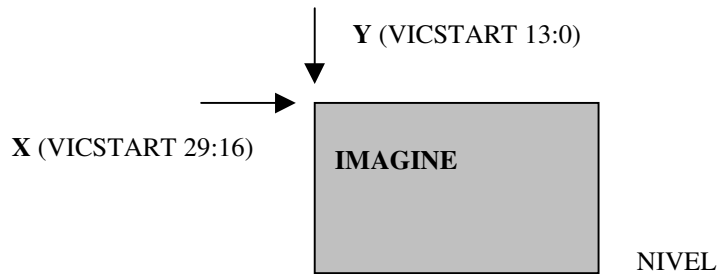


Figura 8.16: coordonatele primului pixel din imagine

VICCTRL stabilește modul color, dacă se lucrează cu dublu tact și dublu port.

VICFCTRL stabilește nivelul și programează parametrii câmpurilor video pare și impare. Se poate face un transfer doar al câmpurilor pare sau impare validând acest lucru în VICFCTRL. Aceasta înseamnă reducerea cantității de date transferate între VPX și SDRAM la jumătate. Un bit din VICFCTRL (SKIP) poate face ca tot al doilea câmp poate să nu fie transferat, ceea ce asociat cu opțiunea de transfer a câmpurilor pare sau impare reduce datele transferate de 4 ori. Un bit (FRAME) stabilește dacă într-un nivel se salvează un cadru (2 câmpuri, unul par și unul impar) sau un singur câmp. În al doilea caz, la afișare liniile sunt dublate pentru a se obține o imagine completă.

VICPCTR stabilește polaritățile semnalelor în modul VPX.

8.2.6. Procesorul grafic GPU

Schema bloc a procesorului grafic este dată în figura 8.17:

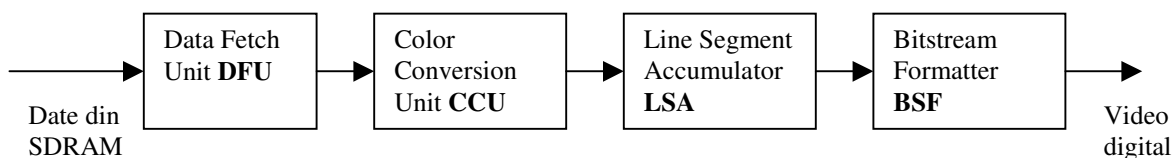


Figura 8.17: schema bloc a procesorului grafic

DFU (Data Fetching Unit) interacționează cu controllerul SDRAM (SDC) pentru a transfera date din memoria RAM video în pipeline. DFU conține un buffer FIFO de 4K biți pentru a uniformiza transferul datelor (memoria RAM video fiind accesată din mai multe locuri). Date (pixeli) din nivele diferite afișate simultan (imagini suprapuse) sunt extrase secvențial din memorie și sunt puse în FIFO. De acolo datele sunt transmise către CCU și LSA.

CCU (Color Conversion Unit) convertește spațiul de culoare al nivelelor într-un spațiu de culoare comun. Modulul de conversie a culorilor conține o tabelă de conversie de 521 cuvinte cu 24 de biți numită CLUT (Colour Look Up Table). Datele pot trece prin CLUT sau nu. CLUT este folosită pentru a converti nivelele cu adâncime de culoare mică în adâncime de culoare mare. Există un CLUT pentru toate nivelele dar se pot defini parametrii diferiți de transformare pentru fiecare nivel. În CCU se pot transforma formatele YUV în RGB și se pot realiza corecții gamma neliniare. Modulatorul DRM (Duty Ratio Modulator) asigură afișarea cu nuanțe de gri sau culoare (pseudonuanțe) pentru datele cu puțini biți de culoare.

LSA (Line Segment Accumulator) realizează ordinea nivelelor în afișare (până la 4 nivele). LSA este de fapt o memorie RAM organizată ca buffer FIFO. LSA are și rolul de a asigura tranziția fluxului de date care sunt prelucrate în GPU cu tactul GPU dar trebuie ca la ieșire să aibă tactul de pixel. Datele din nivelele care se afișează sunt scrise secvențial în LSA de la planul de jos (spate) înspre planurile de sus (față).

BSE (Bit Stream Formatter) pregătește datele pentru a fi trimise spre display în mod digital sau analogic și asigură semnalele de sincronizare.

Observația 1: caracteristica gamma este o relație neliniară între luminozitatea codată în sistemele de televiziune și luminozitatea imaginii. Pași egali în luminozitatea codată nu corespund la pași egali în luminozitatea imaginii. Negrul (luminozitate 0) și albul (luminozitate 1) nu sunt afectate de caracteristica gamma. De exemplu la un tub CRT un semnal video cu luminozitatea 0,5 este afișat cu luminozitatea de 0,21. Pentru a compensa acest efect se realizează corecția gamma care înseamnă modificarea luminozității semnalului video astfel încât să fie afișate nuanțele corecte. În general fiecare culoare R, G și B au nevoie de factori gamma de corecție diferiți. (www.anoca.org/signal/intensity/gamma_correction.htm)

Observația 2: conversia YUV RGB este implementată în Jasmine. Lavender ignoră componentele de culoare din YUV și realizează conversia YUV în 256 de nuanțe de gri (transformare acromatică).

Observația 3. modulatorul DRM asigură nivelele suplimentare de culoare sau de gri pentru a mări adâncimea de culoare. De exemplu, presupunem o frecvență de cadre de 100Hz și un display alb negru (un bit de culoare). Nivelele adiționale pot fi obținute dacă pixelii nu sunt definiți cu culoarea neagră tot timpul, ci doar o parte din timpul total de afișare, ceea ce înseamnă că utilizatorul vede afișată o nuanță de gri. Cu cât pixelii definiți cu culoarea neagră sunt afișați în mai puține cadre, cu atât imaginea este mai deschisă la culoare. Semnalul de sincro V este folosit la modulare, deci gradul de modulare este același pentru toți pixelii. Se definește fiecare pixel cu alb sau negru ca să se obțină nuanțele dorite pe ecran.

Conceptul de spațiu al culorilor

În GDC un pixel se stochează în memoria video RAM cu diferite formate, alocându-se 2, 4, 8, 16 sau 24 de biți pentru un pixel. Formatul de culoare poate fi diferit pentru nivele (plane) diferite. Spațiul de culoare definit de acești biți se numește spațiu logic de culoare. Monitoarele pe care se afișează imaginea pot avea adâncimi de culoare diferite de cele definite în spațiul logic, de aceea spațiul afișabil se numește spațiu fizic de culoare. Spațiul logic trebuie făcut să se potrivească cu spațiul fizic, de aceea s-au integrat în GDC blocuri ca și CLUT, matricea de conversie YUV- RGB și modulatorul DRM, care realizează o reprezentare intermediară numită spațiu intermediar de culoare, figura 8.18:

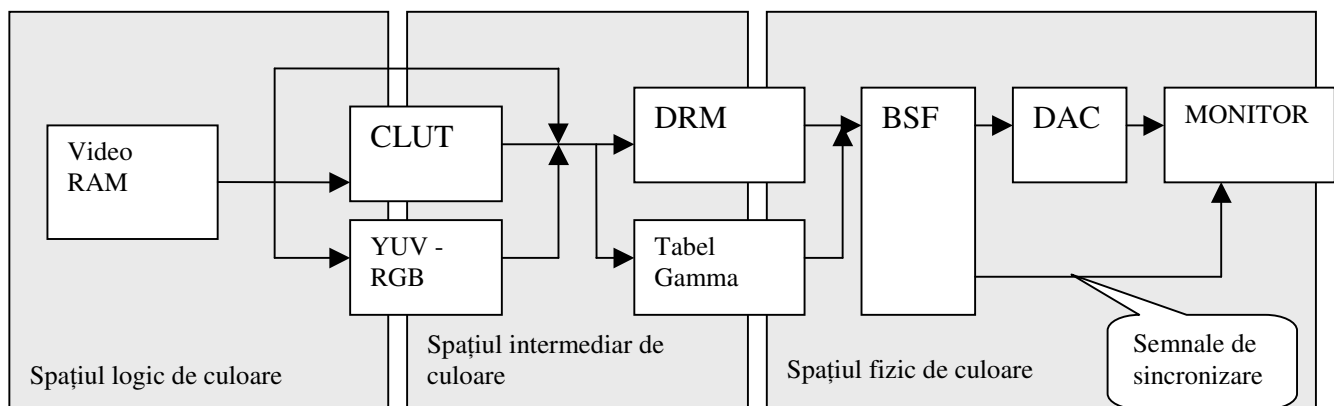


Figura 8.18: schema bloc a procesorului grafic cu punerea în evidență a spațiilor de culoare

De exemplu o conversie din spațiul logic RGB 555 în spațiul intermediar RGB 333 (reducerea adâncimii de culoare) se face prin corespondența (L- logic, I- intermediar):

$I(8:0)=L[14:12]\&L[9:7]\&L[4:2]$, adică au fost aleși biții cei mai semnificativi de culoare din formatul RGB 555 și au fost stocați în noul format intermediar RGB 333 (formatul 15 biți/ pixel a fost transformat în formatul 9 biți/ pixel).

O transformare din YUV 422 în format intermediar RGB necesită transformarea numerică a crominanței printr-o matrice. Transformarea YUV în RGB prin renunțarea la informația de crominanță (în tonuri de gri) se numește acromatică.

Pe display se pot afișa simultan mai multe nivele (domenii grafice sau plane). Reprezentarea acestora în memorie și display este dată în figura 8.19, exemplificând cu 2 plane:

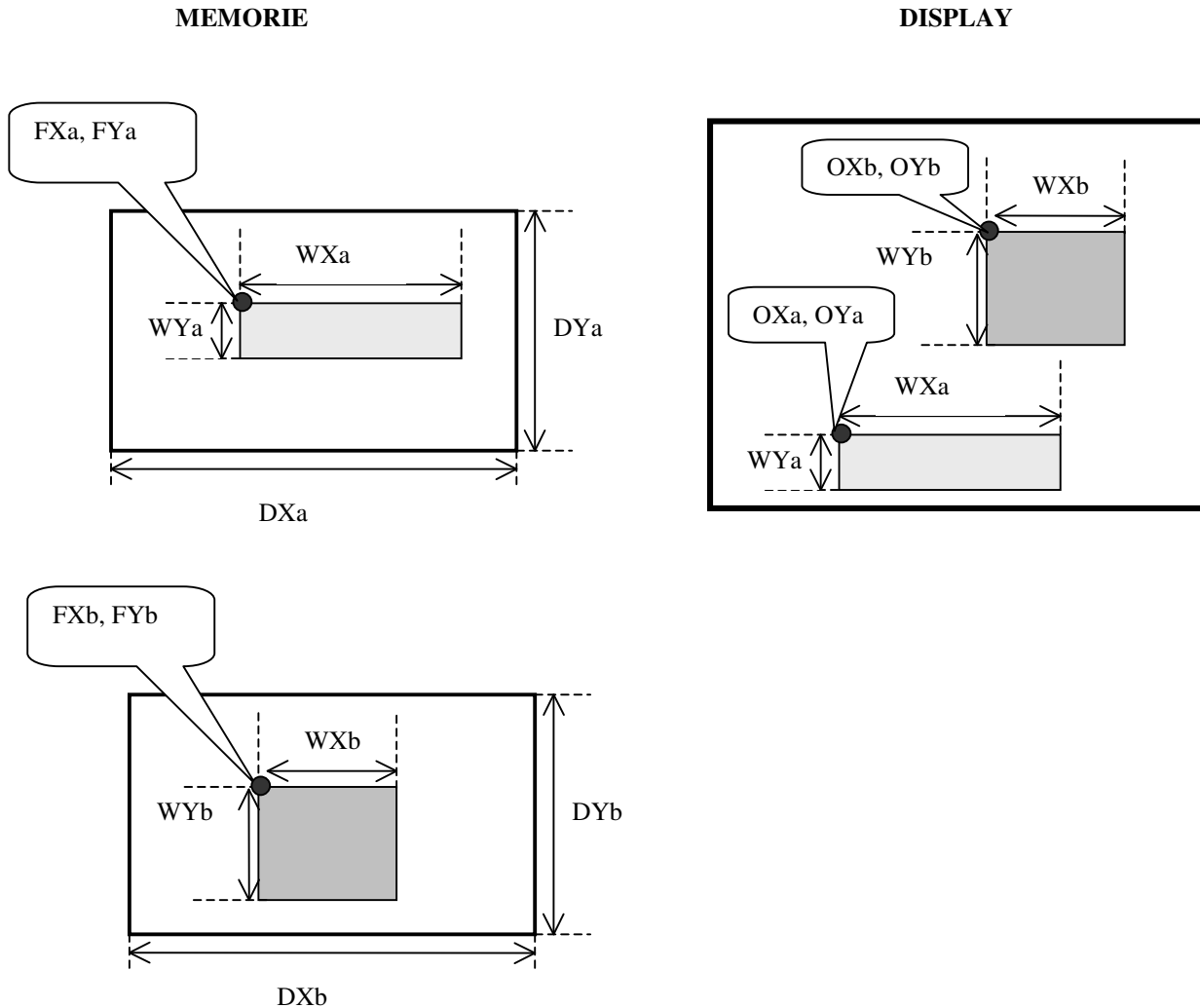


Figura 8.19: afișarea pe mai multe nivele

Domeniul D (cu coordonatele X și Y) reprezintă aria logică acoperită de un nivel în RAM-ul video. FX și FY reprezintă coordonatele primului pixel. WX și WY reprezintă mărimea ferestrei. OX și OY reprezintă offsetul, adică deplasarea nivelului afișat față de colțul din stânga sus a displayului.

Se introduce coordonata z pentru a stabili ordinea nivelelor suprapuse peste bază (background), figura 8.20:

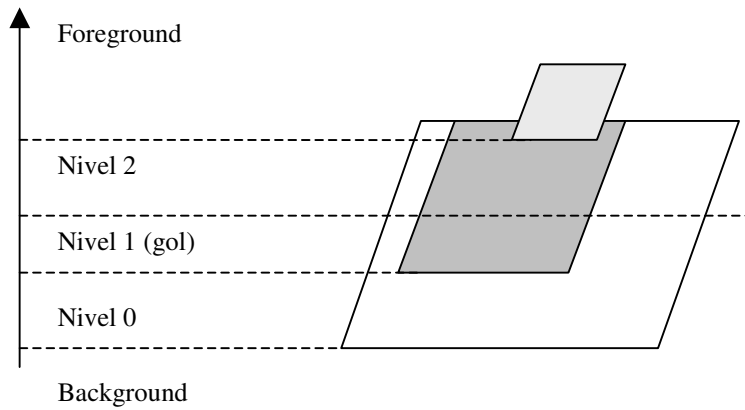


Figura 8.20: coordonata de nivel z

Observația 1

Modul de lucru pereche înseamnă folosirea a 2 display-uri, unul la ieșirea digitală și unul la ieșirea analogică, unul fiind numit display primar și celălalt secundar. Utilizarea a 2 display-uri se poate face definind aceeași rezoluție și aceleași semnale de sincronizare pentru ambele.

Observația 2

Modul de scanare al display-ului (forma în care se trimite șirul de biți la display) poate fi:

- Scanare simplă, pixelii sunt trimiși într-un singur șir.
- Scanare dublă, pixelii sunt trimiși în 2 șiruri paralele. Primul șir este afișat în colțul din stânga sus și parcurge display-ul spre dreapta, linie cu linie. Al doilea șir poate fi afișat începând cu o linie definită printr-un registru și parcurge display-ul la fel.
- Scanare în zig zag, pixelii sunt trimiși într-un singur șir dar liniile se parcurg astfel: prima este 0, a doua este definită printr-un registru +0, a treia este linia 1, a patra este conținutul registrului +1 etc.

8.2.7. Modulul de comandă al afișajelor fluorescente cu catod rece CCFL

Acest modul este folosit pentru a comanda sursa de alimentare în comutație și circuitul de ionizare al afișajului în scopul obținerii unei luminozități ajustabile într-o gamă largă. Schema bloc a sistemului de comandă este dată în figura 8.21:

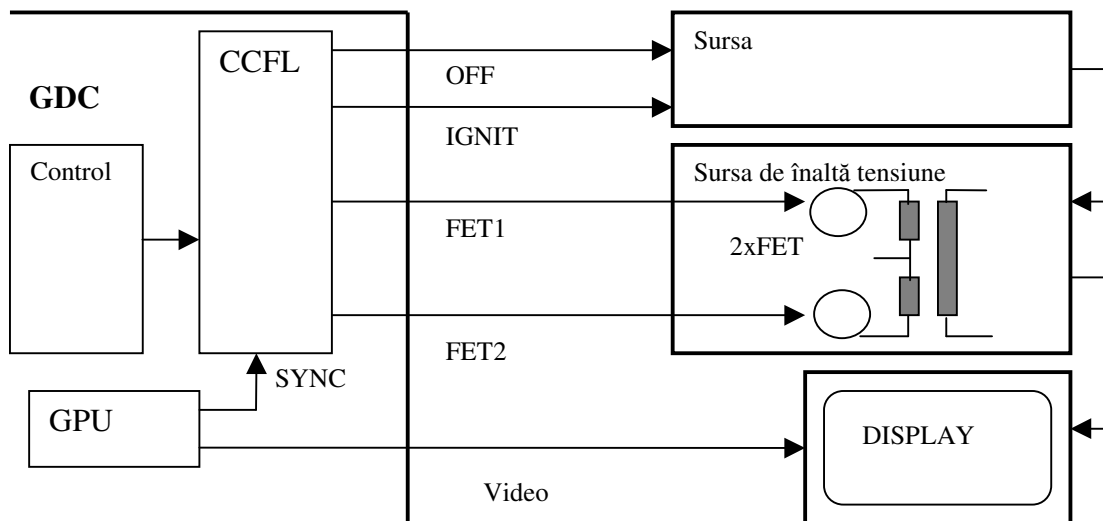


Figura 8.21: schema bloc a modulului de comandă a afișajelor fluorescente

Circuitul CCFL realizează o modulare în durată a iluminării prin controlul tranzistorilor FET complementari din sursa de înaltă tensiune. Semnalele de aprindere (IGNIT) și stingere (OFF) controlează aprinderea și stingerea afișajului în timpul afișării. Modularea iluminării se poate înțelege urmărind diagrama de semnale din figura 8.22:

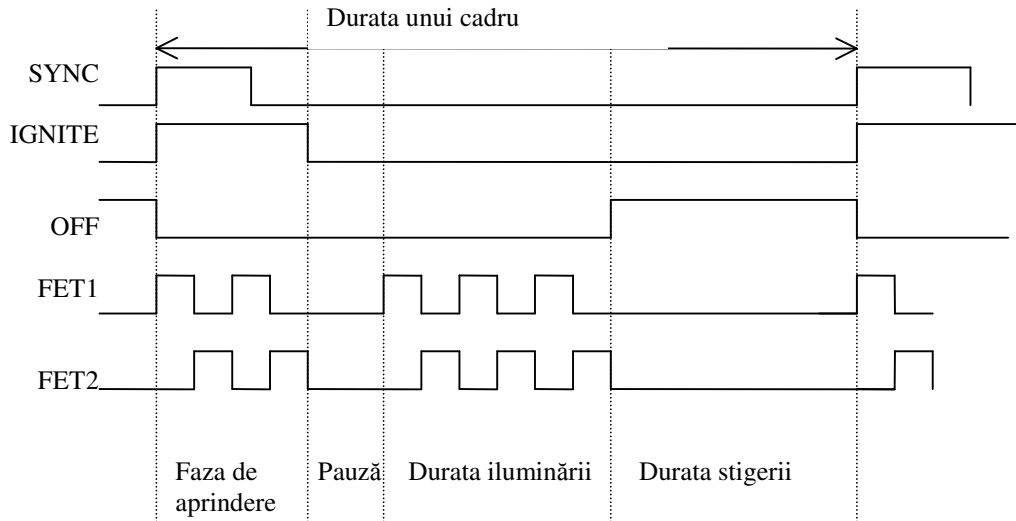


Figura 8.22: diagrame de timp pentru explicarea variației iluminării

Variația iluminării se poate realiza prin modificarea raportului între durata iluminării și durata stingerii.

8.2.8. Cuplearea unui GDC la un microcontroller MB91F361

Circuitul GDC se cuplează la magistrala externă a MC, utilizând o zonă externă de memorie (un bank) și semnalele de CS asociate.

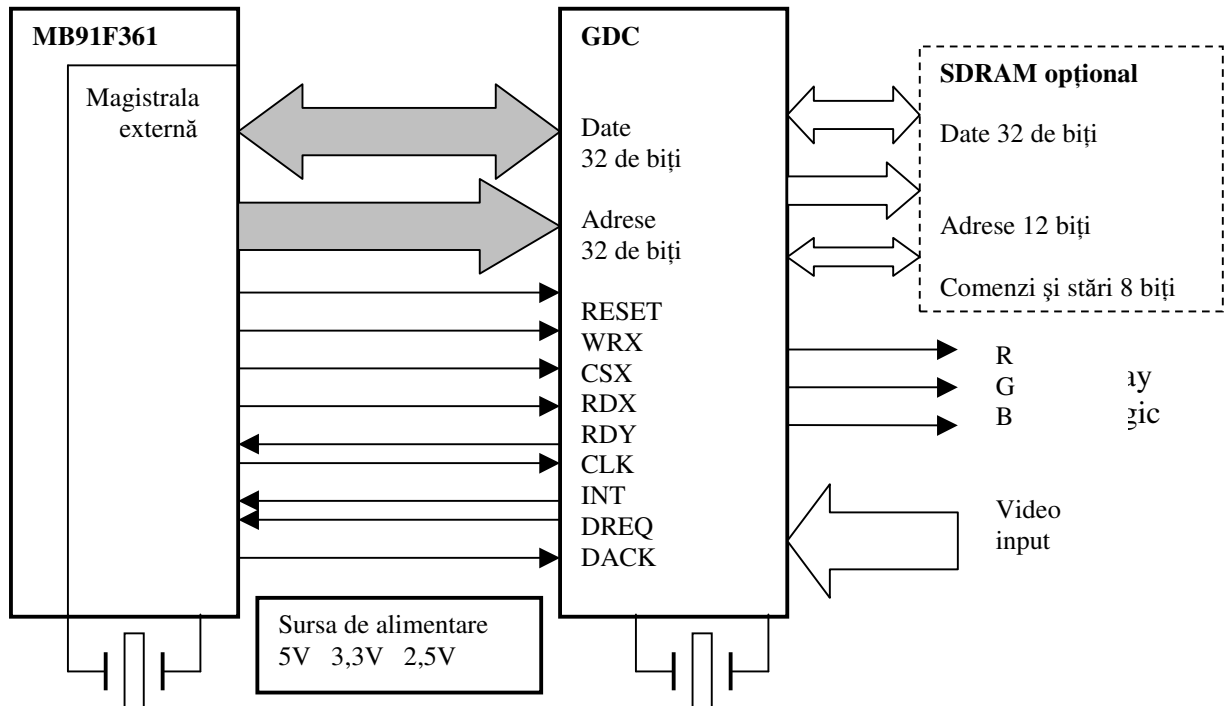


Figura 8.23: cuplarea GDC la un microcontroller MB91F361

Transferul datelor se poate face prin DMA între memoria MC și GDC. Transferurile se fac pe 32 de biți. Schema bloc de conectare demonstrează simplitatea acestei conectări, figura 8.23:

8.2.9. Unelte de dezvoltare

Fujitsu pune la dispoziția utilizatorilor 2 variante de dezvoltare a aplicațiilor cu controllere grafice:

- Placa cu microcontroller pe 32 de biți la care se poate adăuga prin conectori o placă cu controller grafic. Plăcile grafice pot fi echipate cu controllere grafice Cremson, Scarlet, Lavender sau Jasmine, figura 8.24.

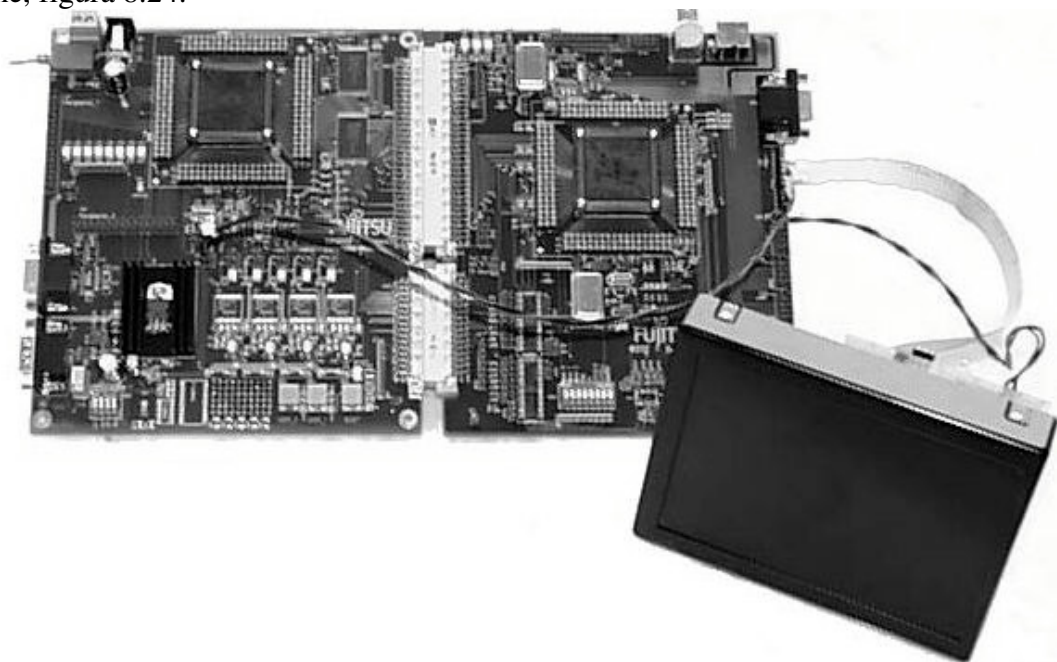


Figura 8.24: placa cu MC pe 32 de biți și placa cu controller grafic

- Placa PCI pentru dezvoltarea de aplicații pe calculatoare PC. Există plăci PCI pentru Cremson, Scarlet, Lavender și Jasmine, figura 8.25:

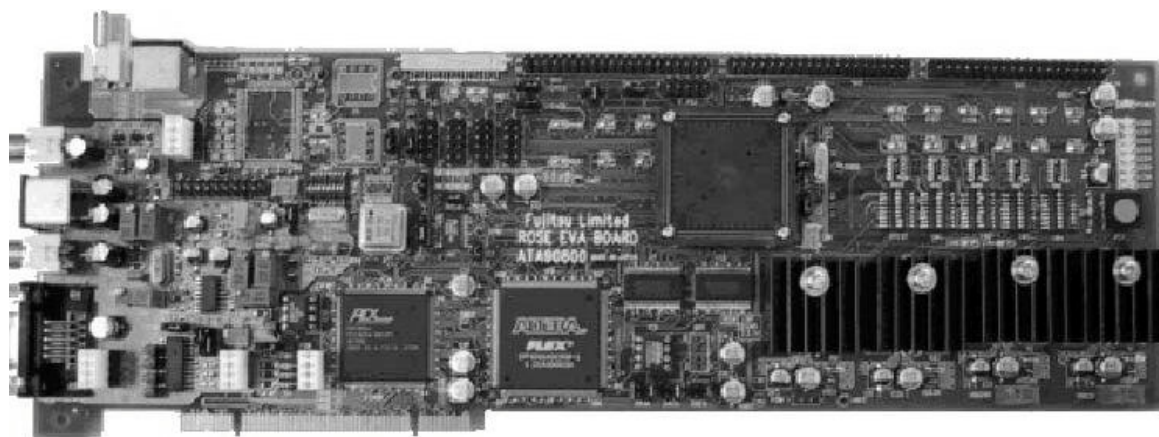


Figura 8.25: placa PCI pentru dezvoltarea de aplicații cu controllere grafice

Din punct de vedere software se pun la dispoziția utilizatorilor drivere grafice în 2 variante:

- Drivere pentru WINDOWS

- Drivere pentru aplicații integrate (embedded)

Aceste drivere fac legătura între software-ul de aplicație și GDC.

Fereastra grafică a unui software simplu care, utilizând driverele grafice, trimite prin interfața serială spre microcontrollerul pe 32 de biți instrucțiunile pentru controllerul grafic este, figura 8.26:



Figura 8.26: fereastra grafică a unui program de trimitere a instrucțiunilor pentru controllerul grafic

După stabilirea parametrilor comunicației, controllerul grafic răspunde:

Lavender Starterkit API Commander Interpreter V1.2

GDC Initialze...OK

>_ (se afișează prompterul interpretorului)

Câteva exemple de instrucțiuni trimise către GDC sunt:

DrawRect 0,0,0,639,479,4 ceea ce înseamnă desenarea unui dreptunghi pe nivelul 0, coordonatele de început (0 pe X, 0 pe Y), coordonatele de sfârșit și numărul de culori (4).

DrawText 0,10,350,"VGA",4 ceea ce înseamnă scrierea textului dintre ghilimele pe nivelul 0, începând cu adresa specificată, cu numărul de culori specificat.

Cele 2 instrucțiuni vor avea ca efect desenarea pe afișaj a următoarei imagini (imagine definită la rezoluția 1024x768), figura 8.27:

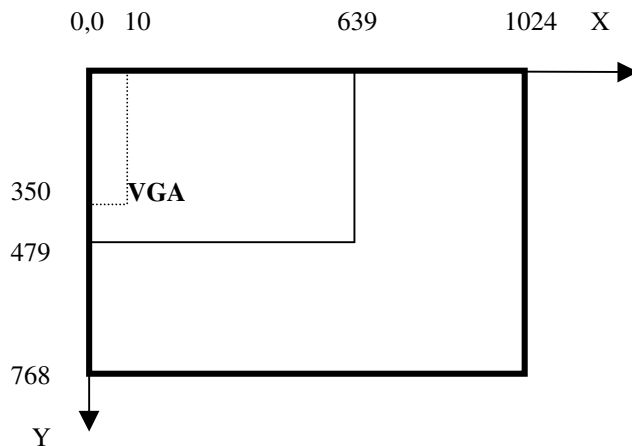


Figura 8.27: desenul care apare pe afișaj ca urmare a instrucțiunilor grafice trimise

9.COMPANIA FUJITSU ȘI POLUAREA

Este bine cunoscută influența negativă pe care o are industrializarea asupra mediului și din păcate foarte multe exemple actuale confirmă această influență. La sfârșitul materialului dedicat microcontrollerelor și controllerelor grafice Fujitsu se menționează câteva exemple care evidențiază tendința companiilor de a conștientiza efectul negativ pe care îl au asupra mediului.

O imagine limpede a importanței acordate mediului în firma Fujitsu poate fi dată analizând structura de conducere, figura 9.1, unde se vede că mediul are aceeași importanță cu activitatea productivă.

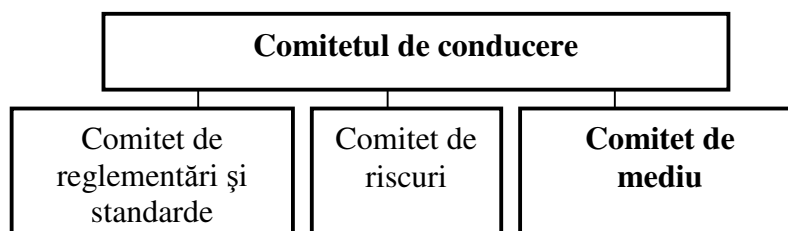


Figura 9.1: structura de conducere a firmei Fujitsu (detaliu)

9.1.Filozofia grupului Fujitsu în domeniul mediului

Principiile în domeniul protecției mediului se bazează pe folosirea cunoștințelor tehnice acumulate în industria IT și a talentului și creativității angajaților pentru a contribui la o dezvoltare durabilă. În activitatea de afaceri se urmăresc și se respectă standardele proprii ale țărilor (în special directiva RoHS) dar se iau și inițiative proprii de protecție a mediului. Prin acțiuni colective și individuale se acționează pentru păstrarea mediului. Principiile Fujitsu de protejare a mediului sunt:

1. Interesul ca produsele firmei să nu afecteze mediul după terminarea ciclului de viață.
2. Conservarea energiei și a resurselor naturale într-o abordare 3R (reducere, reutilizare, reciclare) pentru a realiza produse nepericuloase pentru mediu (eco-friendly).
3. Se încearcă reducerea riscului de îmbolnăvire a omului sau poluarea mediului cu substanțe chimice deversate sau emise în atmosferă.
4. Prin produsele și soluțiile IT se încearcă sprijinirea clienților pentru îmbunătățirea activității lor din punctul de vedere al impactului negativ asupra mediului.
5. Se fac cunoscute activitățile și produsele companiei pentru ca observațiile și criticile să fie folosite pentru îmbunătățirea impactului asupra mediului.
6. Angajații sunt încurajați să îmbunătățească relația cu mediul, analizând impactul afacerilor și responsabilitatea civică.

9.2. Noutăți în direcția protecției mediului

1. Proiectul de reîmpădurire a zonelor din Tailanda, Vietnam și Malaezia, numit “Fujitsu Group Malaysia Eco-Forest Park”, început în 1998, ajuns la faza a 4-a are ca scop protejarea speciilor locale și refacerea ecosistemului. Astfel s-au replantat 100ha în 3 ani și se planifică mărirea suprafeței replantărilor până în 2007. La această acțiune au fost cooptați studenți din universitățile locale și firme locale. O imagine din timpul lucrului este dată în figura 9.2:

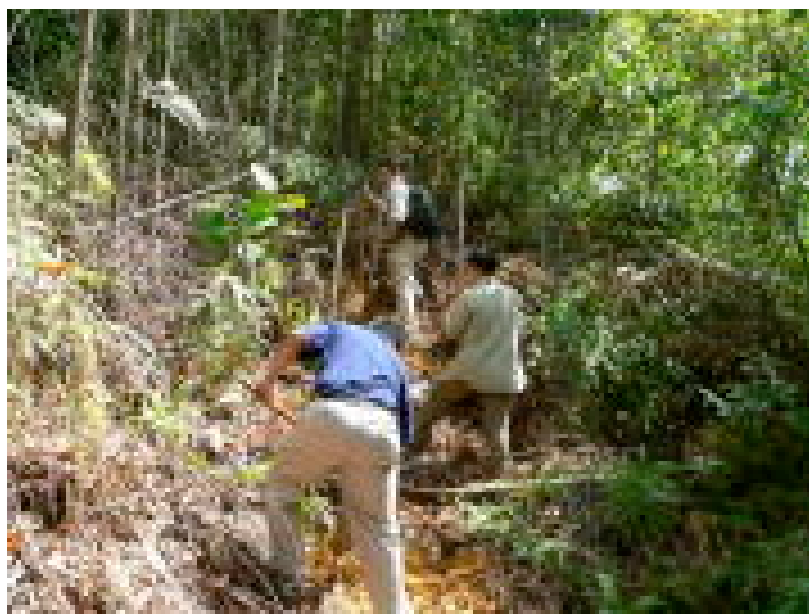


Figura 9.2: imagine din proiectul “Fujitsu Group Malaysia Eco-Forest Park”

2. Fujitsu a hotărât să folosească materiale biodegradabile din plante pentru toate ambalajele de transport ale circuitelor integrate, înlocuind astfel polistirenul. Se estimează că astfel emisiile de bioxid de carbon la fabricarea ambalajelor se va reduce cu 11%, ceea ce se aliniează la cerințele protocolului de la Kyoto ¹ și la legea PRTR ². Materialul folosit este un acid polilactic extras din porumb, care nu produce gaze toxice când este ars. Ambalajele obținute din acest material sunt durabile, oferă protecție bună împotriva descărcărilor electrostatice și își păstrează dimensiunile în timp. Un astfel de ambalaj este prezentat în figura 9.3:

¹ Protocolul de la Kyoto, adoptat în 1992 are ca obiective reducerea emisiilor de gaze care au impact asupra încălzirii globale. Acest protocol, intrat în vigoare în 2005 are ca scop reducerea gazelor emise în 2005-2012 la nivelul celor emise în 1990.

² Legea PRTR (Pollutant Release and Transfer Register) obligă companiile să anunțe cantitățile de gaze emise.

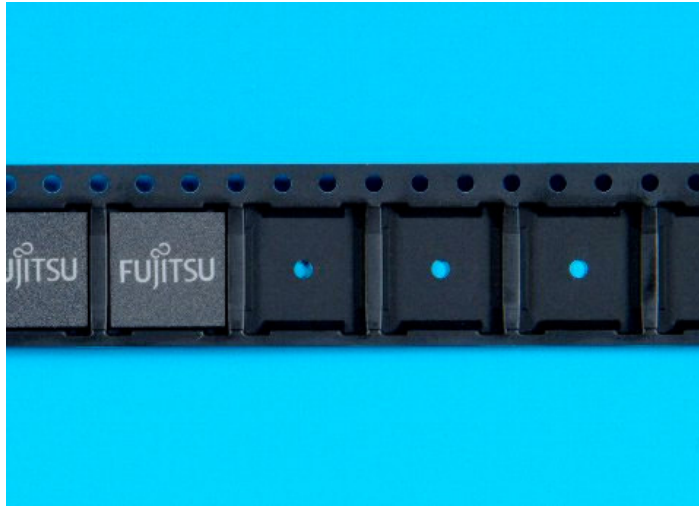


Figura 9.3: ambalaj ecologic

3.În colaborare cu firma TORAY a fost realizat în 2005 o carcasă pentru Notebook-uri realizată din același acid polilactic combinat cu un material plastic, ceea ce oferă caracteristici potrivite pentru o carcasă: duritate, rezistență la încălzire și la flacără etc. Se estimează că emisia de bioxid de carbon va scădea cu 15%. Folosirea acestui material va avea efecte pozitive asupra vânzărilor în ideea legii de promovare a produselor ecologice³.

³ Legea promovării procurării produselor ecologice recomandă cumpărarea produselor cu impact negativ cât mai mic asupra mediului.

BIBLIOGRAFIE

1.Documente Fujitsu

- [1] F²MC-8L MB 89210 Series, Fujitsu Semiconductor Data Sheet, DS07-12558-3E
- [2] F²MC-16LX MB 90350 Series, Fujitsu Semiconductor Data Sheet, DS07-13737-1E
- [3] F²MC-8FX Fujitsu Semiconductor Preliminary
- [4] F²MC-8L MB 89210 Series, Fujitsu Semiconductor Data Sheet, CM25-10150-1E
- [5] FR60Lite MB 91230 Series, Fujitsu Semiconductor Preliminary
- [6] FR50 MB 91360 Series, Fujitsu Semiconductor Data Sheet, CM71-10104-7EA
- [7] MB 87J2120 & MB87P2020, Fujitsu Microelectronics Europe GmbH, MB87P2020fm
- [8] F²MC-16LX MB 90350 Series, Fujitsu Semiconductor Draft-V2-00
- [9] F²MC-16LX MB 90420G/425G Series, Fujitsu Semiconductor Data Sheet, DS07-13711-3E
- [10] 8 & 16 Bit Microcontrollers, Product Overview, Fujitsu Semiconductor
- [11] 32 Bit RISC Microcontrollers, Product Overview, Fujitsu Semiconductor
- [12] Softune Workbench, Getting Started, Application Note, FMEMCU-AN-900025-20 Start SWB 16
- [13] Emulator HW Setup, Application Note FMEMCU-AN-950002-10

2.Referințe INTERNET⁴

europa.eu.int	Directiva ROHS
www.aecouncil.com	AEC-Q100, condiții pentru electronica auto
www.flexray.com	Magistrala Flex Ray
www.anoca.org/signal/intensity/gamma_correction.htm	Corecția gamma- circuite video
www.fujitsu.com	Pagina Fujitsu
http://emea.fujitsu.com/microelectronics	Pagina Fujitsu Europa- noul portal european
www.fme.fujitsu.com	Pagina Fujitsu Europa
www2.rad.com	Ronen Halevi, Udi Nir, Digital Encoding
www.can.bosch.com	Magistrala CAN
www.algonet.se	Tutorial de CAN
www.lin-subbus.org	Specificații Local Interconnect Network LIN
http://www.fujitsu.com/de/services/	Aplicații
http://www.fujitsu.com/us/services/edevices/microelectronics/?grOut=Application+Notes%253fgrOut=Application+Notes&sec=Application%252520Notes	Aplicații
http://www.fdk.co.jp/cyber-e/pi_bz.htm	Aplicații

3.Alte lucrări

- [1] C. Temple, *Protocol Overview*, Flex Ray International Workshop, Detroit 2003

⁴ Unele din aceste referințe au fost menționate și în text