



● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs
embarqués

Compilation pour processeurs
embarqués

Exemple de l'appareil photo
numérique

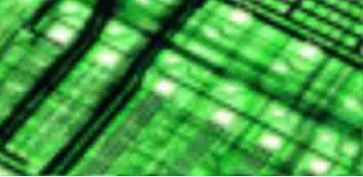
- Exemple de l'appareil photo
numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform:
DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

Exemple de l'appareil photo numérique

Tiré du cours de Franck Vahid:

<http://www.cs.ucr.edu/content/esd/>



Exemple de l'appareil photo numérique

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

Microcontroller seul

● Implémentation 2:

Microcontroller et CCDPP

Conclusion

- Capture les images
- Stocke les images en numérique
 - Éventuellement plusieurs images, le nombre dépend de la taille des images et de la résolution utilisée
- Transmet les images sur PC
- Implementation sur SoC récemment possible:
 - Plusieurs processeurs et mémoires sur un chip
 - Mémoire flash de grande capacité
- Exemple simplifié ici (une seule taille d'image 64×64 , pas de zoom, niveau de gris etc.)

Le point de vue du concepteur

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

Microcontroller seul

● Implémentation 2:

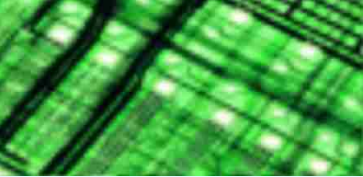
Microcontroller et CCDPP

Conclusion

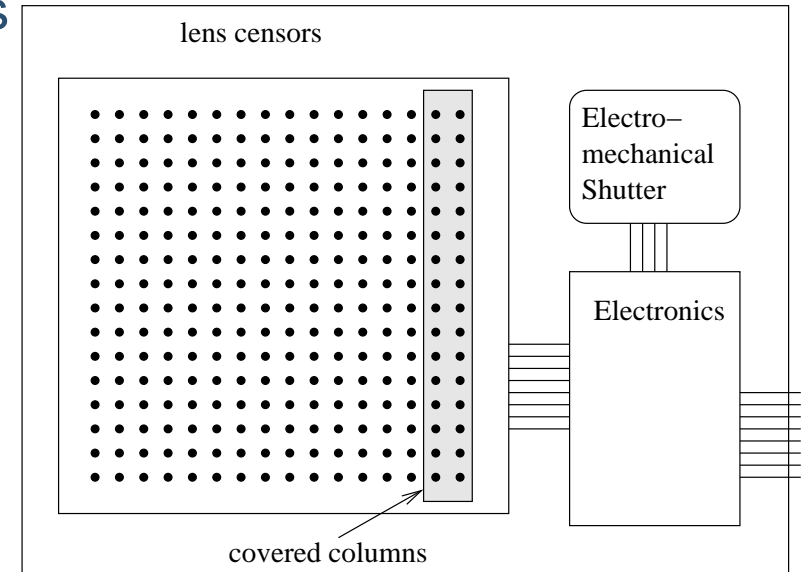
Deux taches importantes:

- Récupérer les images et les stocker dans la mémoire.
Lorsque le déclencheur est pressé:
 - ◆ L'image est capturée par la cellule (CCD)
 - ◆ Elle est convertie en numérique par le CCD
 - ◆ Elle est compressée et stockée dans la mémoire interne
- Transmet une image sur le PC
 - ◆ Connexion série à un PC (UART)

Charge-coupled device (CCD)



- Matrice de senseurs photosensibles (appelées cellules)
- Lorsqu'une cellule est exposée à la lumière elle se charge électriquement, cette charge est convertie en une valeur 8 bits (0=pas d'éclairage)
- Certaines colonnes sont opaque pour servir de référence au 0
- Lorsque que le déclencheur est pressé le circuit électronique décharge les cellules, active le rideau et lit la valeur (8 bits) de chaque cellule
- Ces valeurs sont alors transmises sur un bus pour être stockées en mémoire



● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

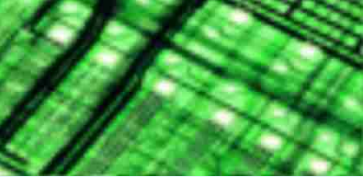
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

Erreur de biais



- Processeurs embarqués

- Introduction

- Architecture des processeurs

- Différents types de processeurs embarqués

- Compilation pour processeurs embarqués

- Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur

- Charge-coupled device (CCD)

- Discrete Cosine Transform: DCT

- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

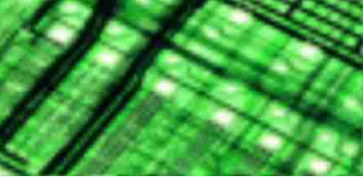
- Conclusion

- Du fait des imperfections, les cellules mesurent un peu plus que la lumière reçue
- Les erreurs sont en général les mêmes sur une même ligne et différentes de ligne en ligne
- Certaines colonnes sont opacifiées pour détecter l'erreur par rapport à 0.
- Si on lit autre chose qu'un zéro sur cette colonne, alors la ligne correspondante doit être mise à jour

cellules non recouvertes								cellules recouvertes		ajustement du zéro
136	170	155	140	144	115	112	248	12	14	
145	146	168	123	120	117	119	147	12	10	-11
144	153	168	117	121	127	118	135	9	9	-9
176	183	161	111	186	130	132	133	0	0	0
144	156	161	133	192	153	138	139	7	7	-7
122	131	128	147	206	151	131	127	2	0	-1
121	155	164	185	254	165	138	129	4	4	-4
173	175	176	183	188	184	117	129	5	5	-5



Erreur de biais



- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique

- Le point de vue du concepteur

- Charge-coupled device (CCD)

- Discrete Cosine Transform: DCT

- Étape de quantization

- Encodage de Huffman

- Contrainte du système

- Spécification fonctionnelle

- Implémentation 1:

Microcontroller seul

- Implémentation 2:

Microcontroller et CCDPP

Conclusion

cellules non recouvertes								cellules recouvertes	
136	170	155	140	144	115	112	248	12	14
145	146	168	123	120	117	119	147	12	10
144	153	168	117	121	127	118	135	9	9
176	183	161	111	186	130	132	133	0	0
144	156	161	133	192	153	138	139	7	7
122	131	128	147	206	151	131	127	2	0
121	155	164	185	254	165	138	129	4	4
173	175	176	183	188	184	117	129	5	5



ajustement du zéro
-13
-11
-9
0
-7
-1
-4
-5



123	157	142	127	131	102	99	235
134	135	157	112	109	106	108	136
135	144	159	108	112	118	109	126
176	183	161	111	186	130	132	133
137	149	154	126	185	146	131	132
121	130	127	146	205	150	130	126
117	151	160	181	250	161	134	125
168	170	171	178	183	179	112	124

Compression

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

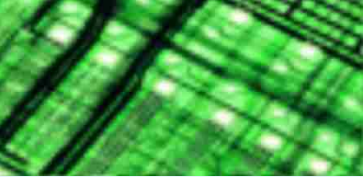
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

- Format JPEG (Joint Photographic Experts Group) pour compresser les images
- Propose différents modes de compression
- Le mode choisi ici permet d'obtenir des taux de compression élevés en utilisant une DCT (Transformé en cosinus discrète)
- L'Image est divisée en blocs de 8×8 pixels
- Trois étapes sur chaque blocs:
 - ◆ DCT
 - ◆ Quantisation
 - ◆ Encodage de Huffman



Discrete Cosine Transform: DCT

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

Microcontroller seul

● Implémentation 2:

Microcontroller et CCDPP

Conclusion

■ Transforme un bloc 8×8 dans un domaine fréquence-cosinus

- ◆ le coin en haut à gauche représente l'essence de l'image
- ◆ Le coin en bas à gauche représente les détails

On peut réduire la précision de cette partie et garder une image de qualité raisonnable

■ Formulation de la FDCT (Forward DCT)

- ◆ $C(h) = 1/\sqrt{2}$ si $h == 0$, 1.0 sinon

fonction auxiliaire utilisée dans la fonction $F(u, v)$

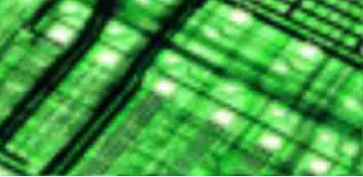
◆

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 D_{xy} \cos(\pi(2x+1)u/16) \cos(\pi(2y+1)v/16)$$

Donne le pixel encodé ligne u colonne v

D_{xy} est la valeur originale du pixel ligne x colonne y

Étape de quantization



● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

Microcontroller seul

● Implémentation 2:

Microcontroller et CCDPP

Conclusion

■ Obtention d'un taux de compression important en diminuant la qualité de l'image

- ◆ Réduire le nombre de bits utilisés pour les données encodées
- ◆ diviser tous les coefficients par une puissance de deux (décalage vers la droite).
- ◆ exemple: division par 8 de chaque cellule après DCT

Après codage DCT

1150	39	-43	-10	26	-83	11	41
-81	-3	115	-73	-6	-2	22	-5
14	-11	1	-42	26	-3	17	-38
2	-61	-13	-12	36	-23	-18	5
44	13	37	-4	10	-21	7	-8
6	-11	-9	-4	20	-28	-21	14
-19	-7	21	-6	3	3	12	-21
-5	-13	-11	-17	-4	-1	7	-4

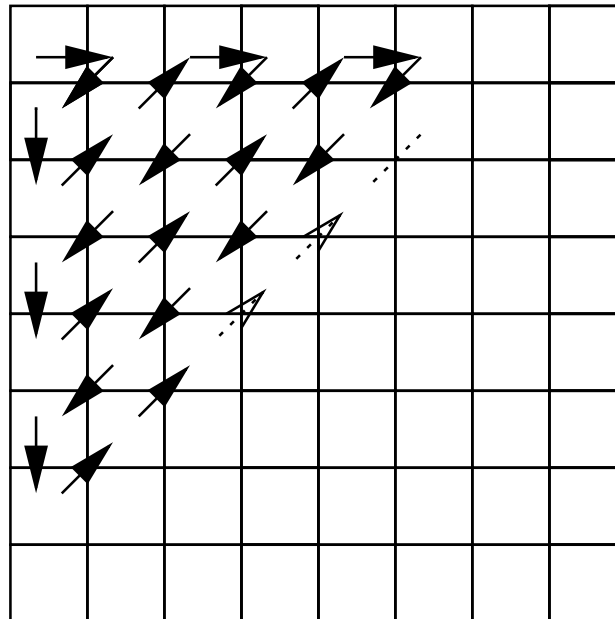
⇒

Après quantization

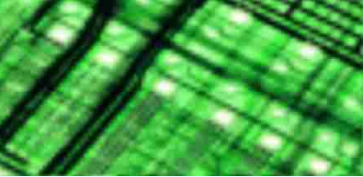
144	5	-5	-1	3	-10	1	5
-10	0	14	-9	-1	0	3	-1
2	-1	0	-5	3	0	2	-5
0	-8	-2	-2	5	-3	-2	1
6	2	5	-1	1	-3	1	-1
5	-1	-1	-1	3	-4	-3	2
-2	-1	3	-1	0	0	2	-3
-1	-2	-1	-2	-1	0	1	-1

Encodage de Huffman

- Sériailisation d'un bloc de 8×8 pixels en utilisant un scan diagonal



- Encodage de Huffman
 - ◆ Aux valeurs les plus fréquentes on assigne un encodage binaire court
 - ◆ Aux valeurs les moins fréquentes on assigne un encodage binaire long



Exemple d'encodage de Huffman

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

Microcontrôleur seul

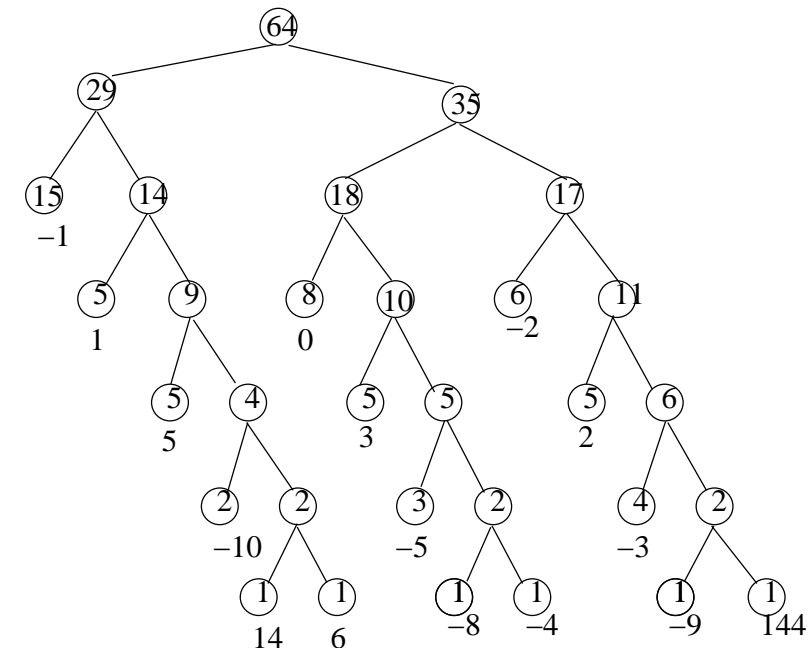
● Implémentation 2:

Microcontrôleur et CCDPP

Conclusion

- Fréquence des pixels: la valeur -1 apparaît 15 fois, la valeur 14 apparaît 1 fois
- Construction de l'arbre de Huffman à partir des valeurs les moins fréquentes
 - ◆ Créer une feuille pour chaque fréquence minimale
 - ◆ combiner par deux les noeuds dont la somme des fréquences est minimale
 - ◆ recommencer jusqu'à avoir intégré toutes les valeurs dans l'arbre binaire.

- Pour obtenir le code de Huffman d'un noeud, atteindre le noeud depuis la racine en marquant 0 pour un fils gauche et 1 pour un fils droit.



valeur	-1	0	-2	1	2	3	5	-3	-5	-10	144	-9	
# apparition	15	8	6	5	5	5	5	4	3	2	1	1	
Huffman code	00	100	110	010	1110	1010	0110	11110	10110	01110	111111	111110	

Chargement de l'image sur un PC

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Lorsque l'appareil est connecté à un PC et qu'il reçoit une commande de chargement
- Lit les images en mémoire
- Les transmet en série sur une interface UART
- Libère la mémoire de l'image chargée

Contrainte du système

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- **Contrainte du système**
- Spécification fonctionnelle
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

- Contraintes non fonctionnelles
Coût, consommation, performances, etc.
- Contraintes fonctionnelles
taux de compression, calcul effectués, etc.
- La spécification initiale peut être très vague.
 - ◆ Prend au moins 50 photos (faible résolution) et les transmet sur un PC
 - ◆ Coûte moins de 100 Euros, contenant un circuit de taille moyenne coûtant moins de 25 Euros.
 - ◆ Batterie durant le plus longtemps possible
 - ◆ Modèle économique de vente.

Contraintes non-fonctionnelles

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- **Contrainte du système**
- Spécification fonctionnelle
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

■ Performances

Approximativement une image par seconde

■ Taille: un chip de moins de 200 000 portes

■ Puissance: le chips ne peut pas être refroidit (ne doit pas dépasser une certaine température)

■ Énergie: consommation minimal possible

■ Ventes espérées :

200 000 si l'arrivée sur le marché est avant 6 mois.

100 000 si l'arrivée sur le marché est entre 6 mois et 1

an

Nul si l'arrivée sur le marché est après 1 an

Spécification fonctionnelle

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

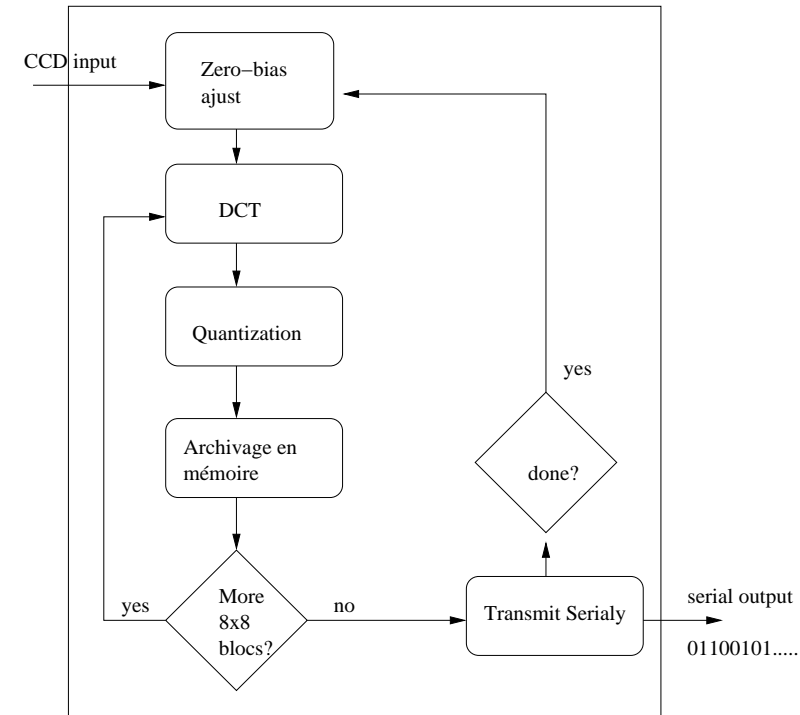
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Spécification souvent informelle (langage naturel), ou UML
- Une image de faible résolution sera 64×64 pixels de 8 bits
- Le mapping des tâches sur le matériel n'est pas encore fait



Spécification fonctionnelle raffinée

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

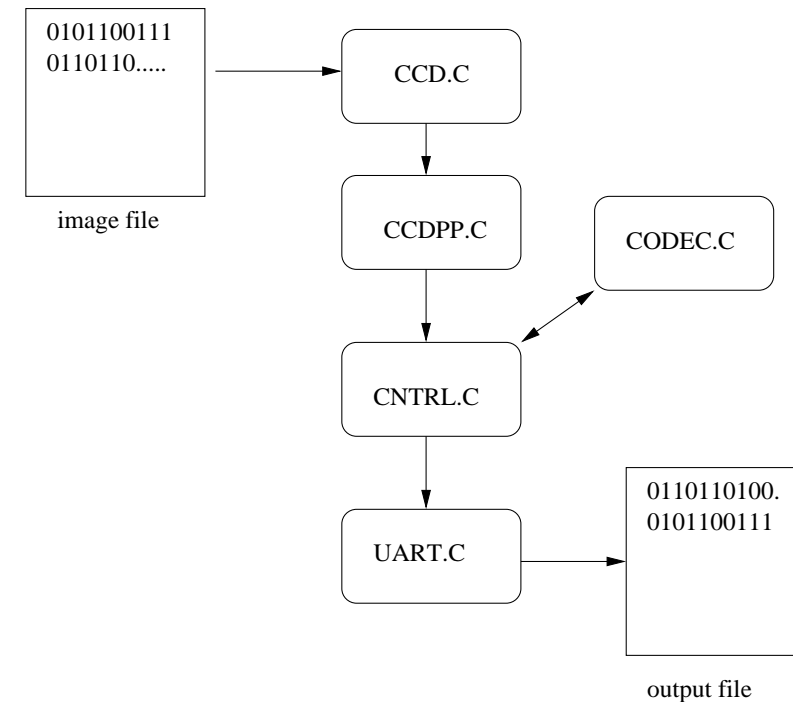
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- **Spécification fonctionnelle**
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Produire une spécification exécutable
- C/C++ souvent utilisé: modèle de niveau système (system level modeling)
- Aide au processus d'implémentation matériel: profilage de l'application
- Produit des données pour vérifier la validité de l'implémentation finale



Module CCD

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- **Spécification fonctionnelle**
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Simule le comportement d'un CCD
- CcdInitialize prend un nom de fichier en argument
- CcdCapture lit une image dans un fichier
- CcdPopPixel envoie les pixels un par un

```
#include <stdio.h>
#define SZ_ROW      64
#define SZ_COL      (64 + 2)
static FILE *imageFileHandle;
static char buffer[SZ_ROW][SZ_COL];
static unsigned rowIndex, colIndex;
```

```
char CcdPopPixel(void) {
    char pixel;
    pixel = buffer[rowIndex][colIndex];
    if( ++colIndex == SZ_COL ) {
        colIndex = 0;
        if( ++rowIndex == SZ_ROW ) {
            colIndex = -1;
            rowIndex = -1;
        }
    }
    return pixel;
}
```

```
void CcdInitialize(const char *imageFileName) {
    imageFileHandle = fopen(imageFileName, "r");
    rowIndex = -1;
    colIndex = -1;
```

```
void CcdCapture(void) {
    int pixel;
    rewind(imageFileHandle);
    for(rowIndex=0; rowIndex<SZ_ROW; rowIndex++) {
        for(colIndex=0; colIndex<SZ_COL; colIndex++) {
            if( fscanf(imageFileHandle, "%i", &pixel) == 1 ) {
                buffer[rowIndex][colIndex] = (char)pixel;
            }
        }
    }
    rowIndex = 0;
    colIndex = 0;
}
```

Module CCDPP (CCD PreProcessing)

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- **Spécification fonctionnelle**
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

- Effectue l'ajustement du zéro
- CcdppCapture utilise CcdCapture et CcdPopPixel pour obtenir l'image.
- effectue l'ajustement après la lecture de chaque ligne

```
void CcdppCapture(void) {
    char bias;
    CcdCapture();
    for(rowIndex=0; rowIndex<SZ_ROW; rowIndex++) {
        for(colIndex=0; colIndex<SZ_COL; colIndex++) {
            buffer[rowIndex][colIndex] = CcdPopPixel();
        }
        bias = (CcdPopPixel() + CcdPopPixel()) / 2;
        for(colIndex=0; colIndex<SZ_COL; colIndex++) {
            buffer[rowIndex][colIndex] -= bias;
        }
    }
    rowIndex = 0;
    colIndex = 0;
}
```

```
#define SZ_ROW      64
#define SZ_COL      64
static char buffer[SZ_ROW][SZ_COL];
static unsigned rowIndex, colIndex;
```

```
void CcdppInitialize() {
    rowIndex = -1;
    colIndex = -1;
}
```

```
char CcdppPopPixel(void) {
    char pixel;
    pixel = buffer[rowIndex][colIndex];
    if( ++colIndex == SZ_COL ) {
        colIndex = 0;
        if( ++rowIndex == SZ_ROW ) {
            colIndex = -1;
            rowIndex = -1;
        }
    }
    return pixel;
}
```

Module UART

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- **Spécification fonctionnelle**
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- On va faire une demi UART (seulement transmission: UAT)
- `UartInitialize` prend le nom du fichier où on va écrire le résultat
- `UartSend` transmet (écrit dans le fichier) octets par octets

```
#include <stdio.h>
static FILE *outputFileHandle;
void UartInitialize(const char *outputFileName) {
    outputFileHandle = fopen(outputFileName, "w");
}
void UartSend(char d) {
    fprintf(outputFileHandle, "%i\n", (int)d);
}
```

Module Codec

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- **Spécification fonctionnelle**
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Modélise l'encodage FDCT
- `ibuffer` contient le bloc original 8×8
- `obuffer` contient le bloc encodé
- `CodecPushPixel` appelé 64 fois pour remplir `ibuffer` avec le bloc original
- `CodecDoFcdt` appelé une fois sur un bloc 8×8 (prochain transparent)
- `CodecPopPixel` appelé 64 fois pour retirer les 64 pixels du bloc encodé.

```
static short ibuffer[8][8], obuffer[8][8], idx;
```

```
void CodecInitialize(void) { idx = 0; }
```

```
void CodecPushPixel(short p) {  
    if( idx == 64 ) idx = 0;  
    ibuffer[idx / 8][idx % 8] = p; idx++;  
}
```

```
void CodecDoFdct(void) {  
    int x, y;  
    for(x=0; x<8; x++) {  
        for(y=0; y<8; y++)  
            obuffer[x][y] = FDCT(x, y, ibuffer);  
    }  
    idx = 0;  
}
```

```
short CodecPopPixel(void) {  
    short p;  
    if( idx == 64 ) idx = 0;  
    p = obuffer[idx / 8][idx % 8]; idx++;  
    return p;  
}
```

Codec (suite)

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle

- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 D_{xy} \cos(\pi(2x+1)u/16) \cos(\pi(2y+1)v/16)$$

- Implémentation de la formule
- Seulement 64 valeurs possible pour $\cos(\pi(2i+1)j/16)$
 - ◆ utilisation d'une table pour améliorer les performances
 - ◆ valeur flottante multipliée par 32 678 (2^{15}) et arrondie à l'entier le plus proche (valeur entière sur 2 octets)
 - ◆ Calculs en virgule fixe
 - ◆ Dans la fonction FDCT: on déroule la sommation interne (sur y) et on implémente la sommation externe (sur x) avec une boucle for.

```
static short ONE_OVER_SQRT_TWO = 23170;
static double COS(int xy, int uv) {
    return COS_TABLE[xy][uv] / 32768.0;
}
static double C(int h) {
    return h ? 1.0
           : ONE_OVER_SQRT_TWO/32768.0;
}
```

```
static const short COS_TABLE[8][8] = {
    { 32768,  32138,  30273,  27245,  23170,  18204,  12539,
      32768,  27245,  12539,  -6392, -23170, -32138, -30273,
      32768,  18204, -12539, -32138, -23170,   6392,  30273,
      32768,   6392, -30273, -18204,  23170,  27245, -12539,
      32768,  -6392, -30273,  18204,  23170, -27245, -12539,
      32768, -18204, -12539,  32138, -23170,  -6392,  30273,
      32768, -27245,  12539,   6392, -23170,  32138, -30273,
      32768, -32138,  30273, -27245,  23170, -18204,  12539,
    };
```

```
static int FDCT(int u, int v, short img[8][8]) {
    double s[8], r = 0; int x;
    for(x=0; x<8; x++) {
        s[x] = img[x][0] * COS(0, v) + img[x][1] * COS(1, v) +
              img[x][2] * COS(2, v) + img[x][3] * COS(3, v) +
              img[x][4] * COS(4, v) + img[x][5] * COS(5, v) +
              img[x][6] * COS(6, v) + img[x][7] * COS(7, v);
    }
    for(x=0; x<8; x++) r += s[x] * COS(x, u);
    return (short)(r * .25 * C(u) * C(v));
}
```

Module controler

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système

● Spécification fonctionnelle

- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Coeur du système
- CntrlInitialize garde la cohérence avec les autres modules
- CntrlCaptureImage utilise le module CCDPP pour charger une image dans un buffer.
- CntrlCompressImage divise l'image 64×64 en blocs de taille 8×8 effectue la quantization puis la DCT sur chaque bloc en utilisant le CODEC.
- CntrlSendImage transmet l'image encodée en série en utilisant le module UART.

```
#define SZ_ROW          64
#define SZ_COL          64
#define NUM_ROW_BLOCKS (SZ_ROW / 8)
#define NUM_COL_BLOCKS (SZ_COL / 8)
static short buffer[SZ_ROW][SZ_COL],
static short i, j, k, l, temp;
void CntrlInitialize(void) {}
```

```
void CntrlCaptureImage(void) {
    CcdppCapture();
    for(i=0; i<SZ_ROW; i++)
        for(j=0; j<SZ_COL; j++)
            buffer[i][j] = CcdppPopPixel();
}
```

```
void CntrlSendImage(void) {
    for(i=0; i<SZ_ROW; i++)
        for(j=0; j<SZ_COL; j++) {
            temp = buffer[i][j];
            UartSend(((char*)&temp)[0]); /* send upper byte
            UartSend(((char*)&temp)[1]); /* send lower byte
        }
    }
}
```

```
void CntrlCompressImage(void) {
    for(i=0; i<NUM_ROW_BLOCKS; i++)
        for(j=0; j<NUM_COL_BLOCKS; j++) {
            for(k=0; k<8; k++)
                for(l=0; l<8; l++)
                    CodecPushPixel(
                        (char)buffer[i * 8 + k][j * 8 + l]);
            CodecDoFdct(); /* part 1 - FDCT */
            for(k=0; k<8; k++)
                for(l=0; l<8; l++) {
                    buffer[i*8+k][j*8+l] = CodecPopPixel();
                    /* part 2 - quantization */
                    buffer[i*8+k][j*8+l] >>= 6;
                }
        }
    }
}
```

Flot d'exécution

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

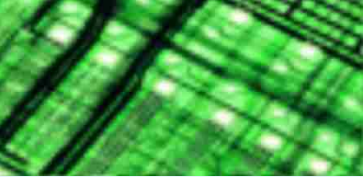
Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1:
Microcontroller seul
- Implémentation 2:
Microcontroller et CCDPP

Conclusion

- `main` initialise chacun des modules puis utilise `CNTRL` pour capturer, compresser et transmettre une image.
- Cette spécification de niveau système est utilisée pour sa rapidité d'exécution et facilité de débogage

```
int main(int argc, char *argv[]) {
    char *uartOutputFileName = argc > 1 ? argv[1] : "uart_out.txt";
    char *imageFileName = argc > 2 ? argv[2] : "image.txt";
    /* initialize the modules */
    UartInitialize(uartOutputFileName);
    CcdInitialize(imageFileName);
    CcdppInitialize();
    CodecInitialize();
    CntrlInitialize();
    /* simulate functionality */
    CntrlCaptureImage();
    CntrlCompressImage();
    CntrlSendImage();
}
```



Implémentation 1: Microcontroller seul

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle

● Implémentation 1: Microcontroller seul

- Implémentation 2: Microcontroller et CCDPP

Conclusion

■ Le designer choisi d'utiliser un microcontroller 8051

- ◆ Le 8051 est un microcontrôleur 8 bit simple basé sur un accumulateur, il possède 255 instructions.
- ◆ Une instruction dure 12 cycles d'horloge (4 dans quelques modèles).
- ◆ Le CPU a quatre bancs de 8 registres 8 bits dans une RAM sur le chip (context switching) qui résident dans les premiers 128 octets de la mémoire avec une région utilisée pour les opérations de niveau bit et une scartchpad memory.

■ Coût total du circuit: 5 Euros

■ Bien en dessous de 200 mW

■ Time-to-market à peu près 3 mois.

■ comment évaluer les performances ?

- ◆ On récupère un compilateur pour 8051, par exemple SDCC (Small device C Compiler)

<http://sdcc.sourceforge.net/>

- ◆ On compile et on regarde

- ◆ éventuellement, on récupère un simulateur du 8051, par exemple PDS-52

<http://www.phyton.com/downloads/project-52.zip>

et on simule l'exécution du code

Implémentation 1: Microcontroller seul

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

■ Les performances ne sont pas suffi santes:

- ◆ 12 MHz, 12 cycles par instructions: 1 millions d'instructions par seconde
- ◆ CcdppCapture à une boucle boucle de 4096 itérations (64 x 64) avec 100 instructions assembleur par itération soit 409,000 (4096 x 100) instructions par image
- ◆ 1000 instructions assembleur pour la sommation $s[x] =$

```
img[x][0] * Y(0, v) + img[x][1] * Y(1, v) + img[x][2] * Y(2, v) + img[x][3] * Y(3, v) +  
img[x][4] * Y(4, v) + img[x][5] * Y(5, v) + img[x][6] * Y(6, v) + img[x][7] * Y(7, v);
```

exécutée $64 \times 8 = 512$ fois, soit 512000 instructions

- ◆ il n'y a plus de temps pour les autres calculs

Implémentation 2: Microcontroller et CCDPP

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

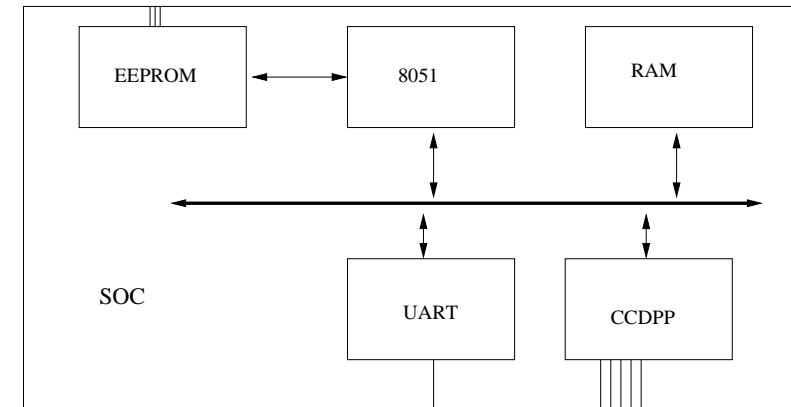
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- **Implémentation 2: Microcontroller et CCDPP**

Conclusion

- La fonction de CCDPP implémenté sur un circuit dédié
- Améliore la performance mais augmente le coût et le temps d'arrivé dur le marché
- Facile a implémenter (chemin de donnée simple avec un controller à quelques états)
- une UART simple, un EEPROM pour le programme du 8051 ainsi que de la mémoire pour stocker les images



Micro-contrôleur synthétisable

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

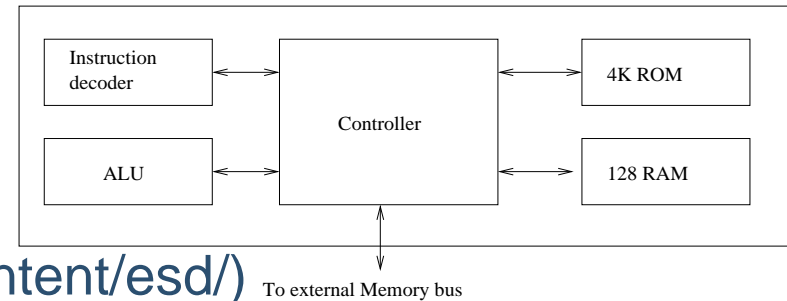
Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

■ Version synthétisable du microprocesseur Intel 8051

(<http://www.cs.ucr.edu/content/esd/>)



■ description de niveau transfer de registre (RTL)

- Charge les instructions depuis la ROM
- Décode avec le décodeur d'instruction
- l'ALU exécute les instructions arithmétiques
- les registres sont placés dans la RAM on-chip .
- instruction spécifi que pour charger et stocker dans la RAM externe
- Un pretty printer permet de générer l'initialisation de la ROM à partir de l'assembleur généré par le compilateur

8051: version plus détaillée

● Processeurs embarqués

Introduction

Architecture des processeurs

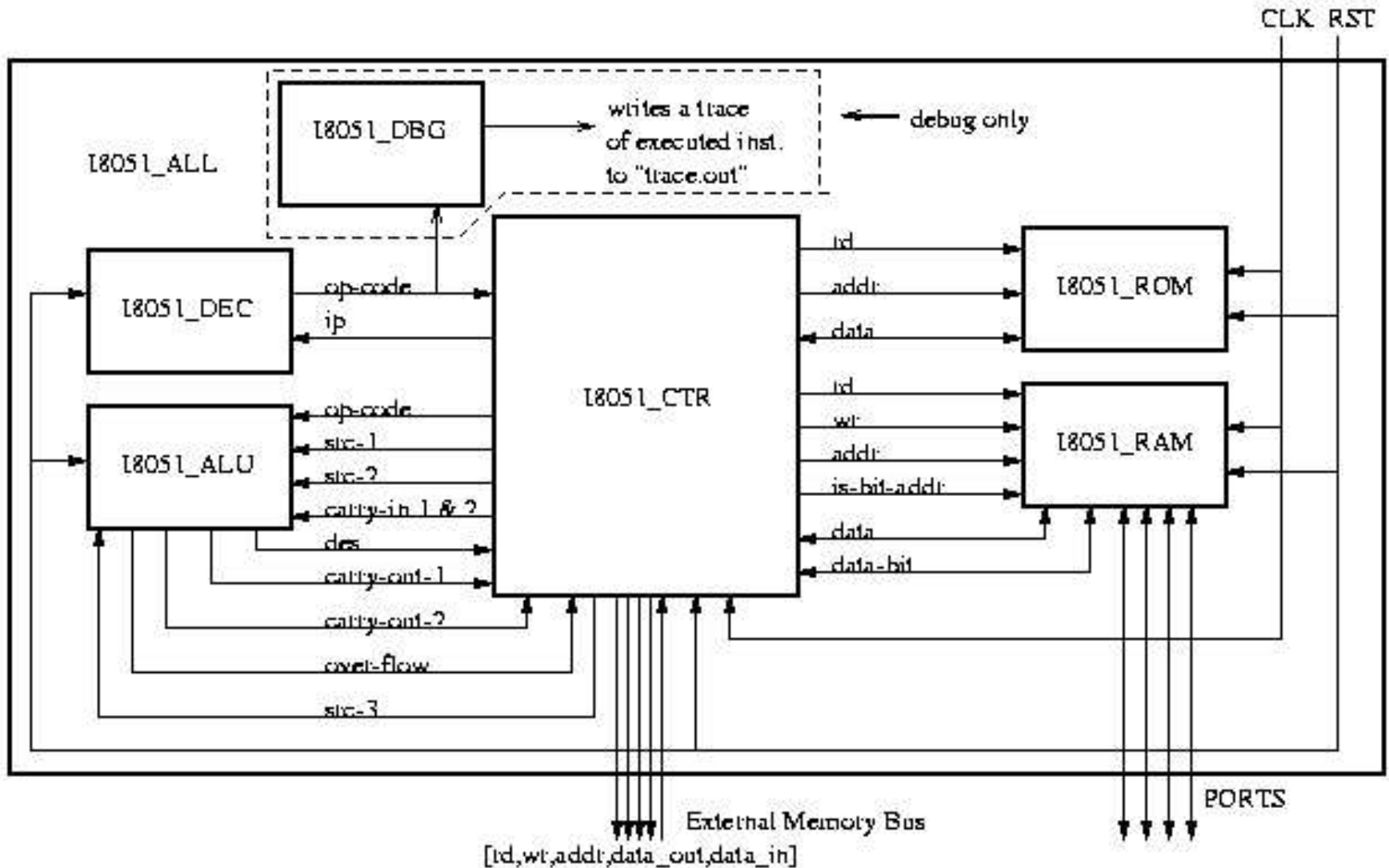
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- **Implémentation 2: Microcontroller et CCDPP**

Conclusion



UART

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

● Exemple de l'appareil photo numérique

● Le point de vue du concepteur

● Charge-coupled device (CCD)

● Discrete Cosine Transform: DCT

● Étape de quantization

● Encodage de Huffman

● Contrainte du système

● Spécification fonctionnelle

● Implémentation 1:

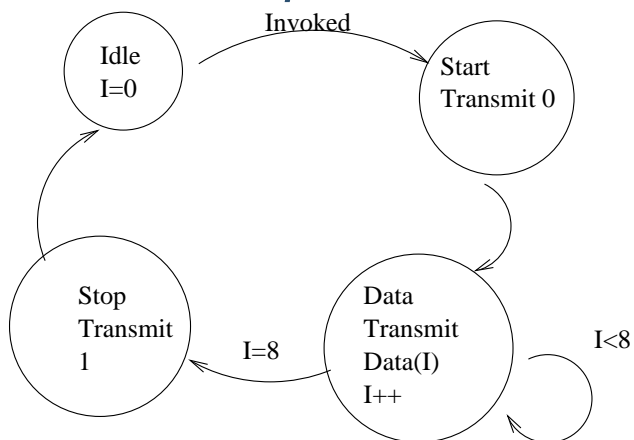
Microcontroller seul

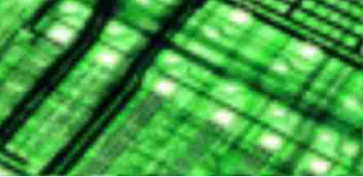
● Implémentation 2:

Microcontroller et CCDPP

Conclusion

- L'UART est invoquée lorsque le 8051 veut stocker une image en écrivant à l'adresse correspondant au registre *enable* de l'UART (Memory mapped I/O).
- L'état *Start*, lorsqu'il est invoqué transmet 0 pour indiquer une nouvelle transmission.
- La donnée est envoyée bit par bit
- L'état *Stop* transmet 1 et l'UART revient dans l'état *Idle*





CCDPP

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

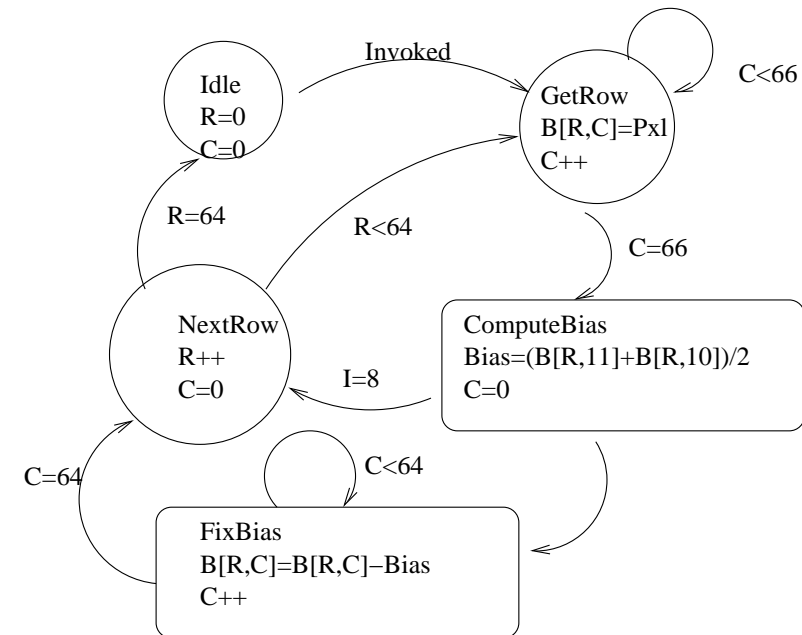
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontrôleur seul
- **Implémentation 2: Microcontrôleur et CCDPP**

Conclusion

- L'implémentation matérielle de la correction du biais interagit avec le chip CCD.
- Contient un buffer particulier pour l'image: B mappé en mémoire du 8051
- les Variables R, C sont les indices des lignes et colonne du pixel adressé dans B
- L'état *GetRow* lit une ligne du CCD dans B (66 pixels).
- l'état *ComputeBias* calcule le biais pour la ligne. l'état *FixBias* corrige le biais.
- *NextRow* ramène à *GetRow* ou dans l'état *Idle* quand les 64 lignes sont lues



Connexion des composants du SoC

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Tous les composants mappé en mémoire du 8051 sont connectés au bus mémoire du 8051.
- Lorsque que le processeur écrit:
 - ◆ il place l'adresse sur le bus de 16 bit
 - ◆ Il valide (*assert*) le signal `read` pendant 1 cycle
 - ◆ Il lit la valeur sur le bus de donnée au cycle suivant
 - ◆ Le périphérique (RAM ou circuit dédié) détecte le signal `Read`, vérifi e l'adresse
 - ◆ place la donnée sur le bus de donnée 8 bits pendant 1 cycle,
- Lorsque le processeur écrit
 - ◆ Il place l'adresse et la donnée sur les bus d'adresse et de donnée
 - ◆ il active le signal `Write` pendant un cycle
 - ◆ Le périphérique (RAM ou circuit dédié) détecte le signal `Write`, vérifi e l'adresse, lit et stocke la donnée du bus de donnée.

Logiciel

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Le model déjà écrit fournit la majorité du code, on garde la même hiérarchie, le code de l'UART et CCDPP doivent être revues.
- remplacement des assignation en mémoire par
 - ◆ le type `xdata` est utilisé pour stocker sur la RAM externe
 - ◆ La directive `at` spécifie l'adresse où doit résider une variable
 - ◆ EX: Un octet envoyé à l'adresse `U_STAT_REG` invoquera l'UART.

Code originale de l'UART

```
#include <stdio.h>
static FILE *outputFileHandle;
void UartInitialize(const char *outputFileName) {
    outputFileHandle = fopen(outputFileName, "w");
}
void UartSend(char d) {
    fprintf(outputFileHandle, "%i\n", (int)d);
}
```

⇒

Code Réécrit de l'UART

```
static unsigned char xdata U_TX_REG_at_ 65535
static unsigned char xdata U_STAT_REG_at_ 655
void UARTInitialize(void) {}
void UARTSend(unsigned char d) {
    while( U_STAT_REG == 1 ) {
        /* busy wait */
    }
    U_TX_REG = d;
}
```


Synthèse et Analyse

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Le SoC complet peut être simulé en VHDL:
- Avant synthèse logique:
 - ◆ Vérification fonctionnelle
 - ◆ Validation des performances
 - ◆ Estimation de la surface
- Après synthèse logique:
 - ◆ Vérification des timing (clock cycle) et performances
- Après placement routage:
 - ◆ Vérification précise des timing
 - ◆ Simulation de la consommation électrique (décompte du nombre de switch entre 0 et 1 à chaque cycles)
 - ◆ Calcul de la surface précise.

Résultat de F. Vahid pour l'implémentation 2

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Temps d'exécution pour traiter une image 9.1 seconds
- Power consumption: 0.033 watt
- Energy consumption: 0.30 joule (9.1 s x 0.033 watt)
- Surface du chip : 98,000 gates

Implémentation 3: 8051, CCDPP et Codec

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

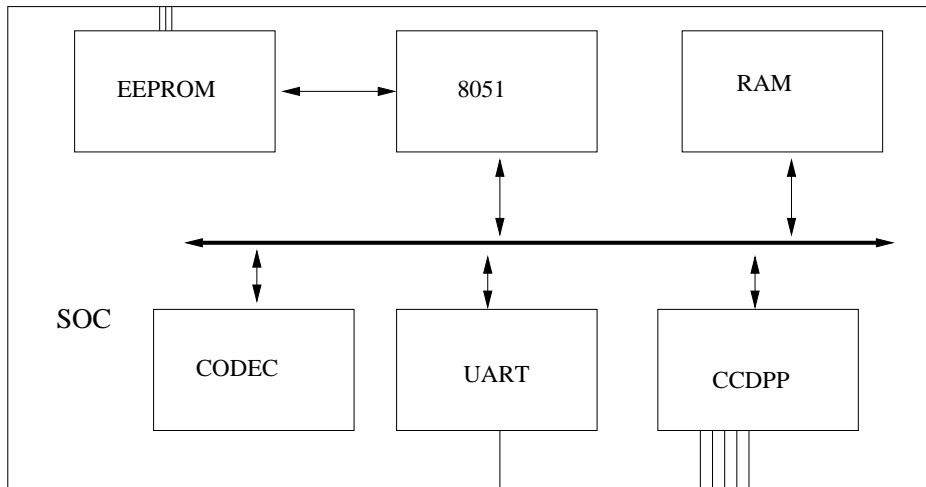
Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- **Implémentation 2: Microcontroller et CCDPP**

Conclusion

- Les performances ne sont pas atteintes
- les simulations montrent que la majorité du temps est passé par le 8051 dans la DCT
- Le codec est donc un candidat au passage en matériel



Conception du CODEC

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- 4 registres memory mapped
 - ◆ C_DATAI_REG / C_DATAO_REG utilisé pour recevoir/envoyer les blocs 8×8 .
 - ◆ C_CMND_REG utilisé pour commander le codec
 - ◆ C_STAT_REG indique que le CODEC a terminé et est prêt pour le bloc suivant. Le 8051 fait du *polling* (teste ce registre en boucle)
- La traduction en matériel consiste essentiellement à passer en virgule fixe.
- La version logicielle du CODEC devient presque vide

```
static unsigned char xdata C_STAT_REG _at_ 65527;
static unsigned char xdata C_CMND_REG _at_ 65528;
static unsigned char xdata C_DATAI_REG _at_ 65529;
static unsigned char xdata C_DATAO_REG _at_ 65530;
void CodecInitialize(void) {}
void CodecPushPixel(short p) { C_DATAO_REG = (char)p; }
short CodecPopPixel(void) {
    return ((C_DATAI_REG << 8) | C_DATAI_REG);
}
void CodecDoFdct(void) {
    C_CMND_REG = 1;
    while( C_STAT_REG == 1 ) { /* busy wait */ }
}
```

Résultat de F. Vahid pour l'implémentation 3

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Temps d'exécution pour traiter une image 0.099 seconds
- Power consumption: 0.040 watt: augmenté car il y a deux processeurs sur le chip
- Energy consumption: 0.0040 joule (0.099 s x 0.040 watt): la vie de la batterie sera 12 fois plus longue que pour l'implémentation précédente
- Surface du chip : 128,000 gates: augmentation significative

Conclusion

● Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

- Le travail essentiel d'un concepteur aujourd'hui est de savoir faire le compromis logiciel/matériel
- Il doit savoir programmer une implémentation logicielle
- Il doit savoir comment accélérer certains traitements par le matériel
- Il doit prévoir au mieux les performances des compromis envisagés pour chaque indicateurs