

Enseignes et afficheurs à LED

# Accès direct en mémoire (DMA)

# Accès direct en mémoire (DMA)

**Pierre-Yves Rochat**

- Limite du débit d'un microcontrôleur
- Architecture du DMA
- Exemple simple
- Application à une matrice de LED

# Limite du débit de sortie d'un microcontrôleur

---

- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées

# Limite du débit de sortie d'un microcontrôleur

---

- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées
- Des microcontrôleurs dont la fréquence du processeur est plus élevés peuvent être utilisés

# Limite du débit de sortie d'un microcontrôleur

---

- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées
- Des microcontrôleurs dont la fréquence du processeur est plus élevés peuvent être utilisés
- La technique de l'accès direct en mémoire repousse un peu la limite

# Limite du débit de sortie d'un microcontrôleur

- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées
  - Des microcontrôleurs dont la fréquence du processeur est plus élevés peuvent être utilisés
  - La technique de l'accès direct en mémoire repousse un peu la limite
- 
- **L'idée** : ne pas à avoir exécuter des instructions pour chaque transfert de donnée

# Limite du débit de sortie d'un microcontrôleur

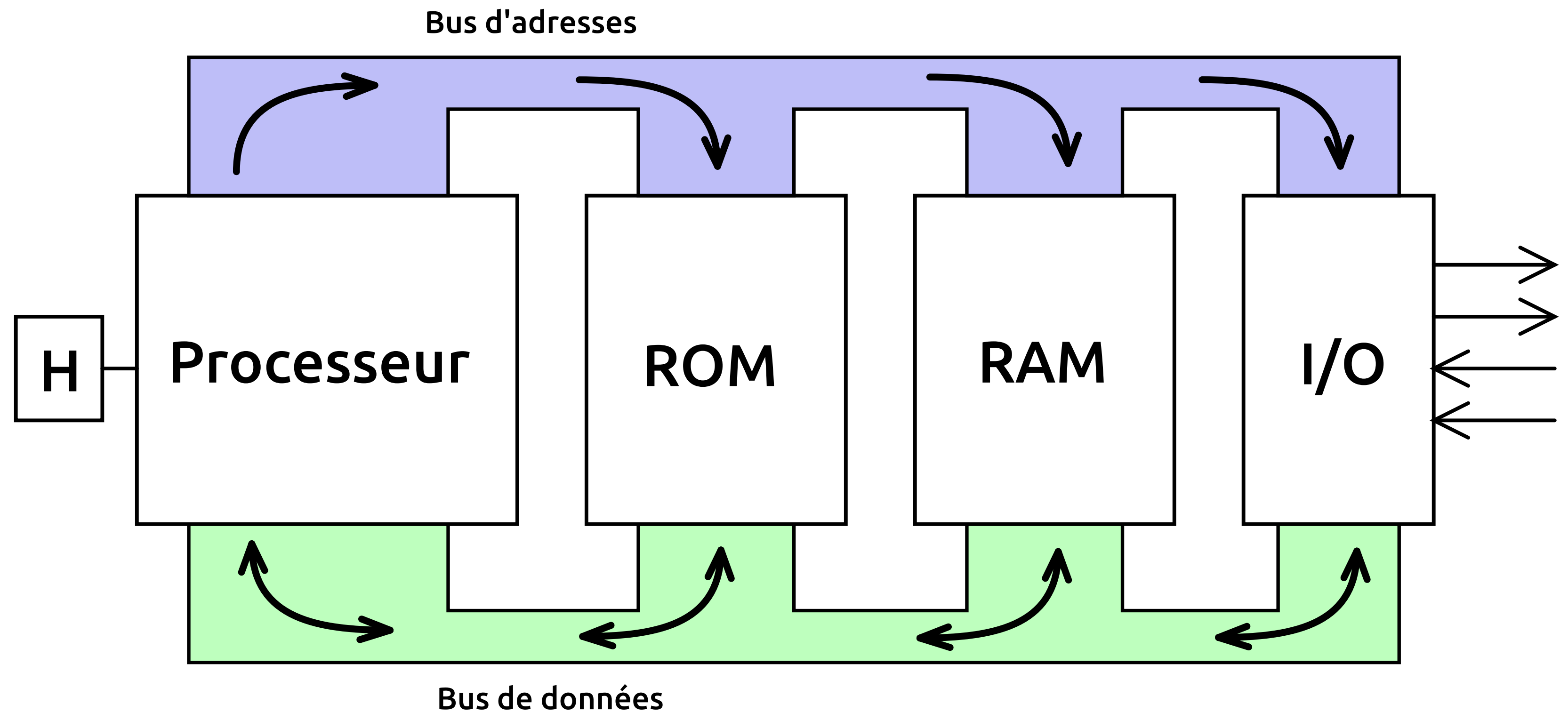
- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées
  - Des microcontrôleurs dont la fréquence du processeur est plus élevés peuvent être utilisés
  - La technique de l'accès direct en mémoire repousse un peu la limite
- 
- **L'idée** : ne pas à avoir exécuter des instructions pour chaque transfert de donnée
  - Elle n'est pas nouvelle : le circuit Intel 8253 date de 1981...

# Limite du débit de sortie d'un microcontrôleur

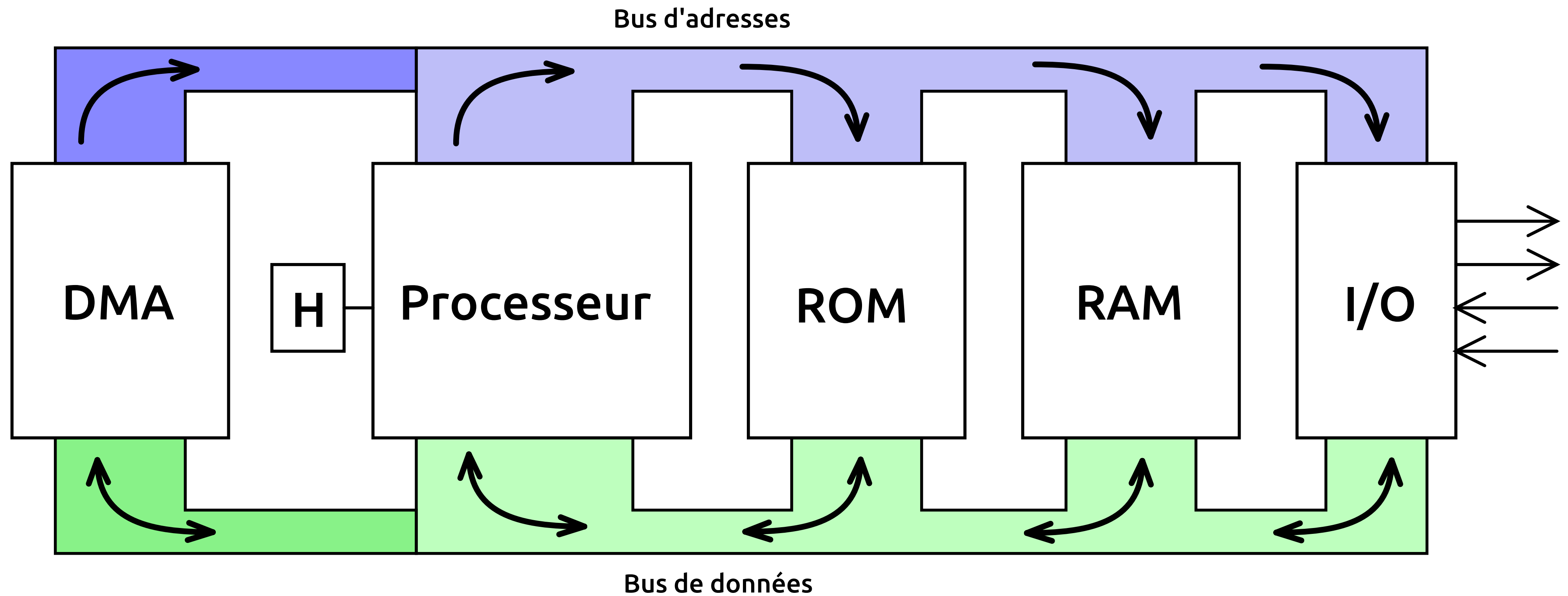
- Les matrices comportant beaucoup de LED exigent des vitesses de rafraîchissement élevées
  - Des microcontrôleurs dont la fréquence du processeur est plus élevés peuvent être utilisés
  - La technique de l'accès direct en mémoire repousse un peu la limite
- 
- **L'idée** : ne pas à avoir exécuter des instructions pour chaque transfert de donnée
  - Elle n'est pas nouvelle : le circuit Intel 8253 date de 1981...
  - et on en retrouve des traces dans les PC récents !



# Architecture d'un système informatique



# Architecture du DMA



- Principe simple...
- Documentation complexe
- Très grand nombre de registres

# Programme le plus simple possible

```
1 int main(void) {  
2     HAL_Init();  
3     InitGPIOA();  
4     InitTame();  
5     InitDMA();  
6     InitTimer1();  
7  
8     while(1) {  
9     }  
10 }
```

# Initialisation du DMA

```

1 void InitDMA() {
2     __HAL_RCC_DMA2_CLK_ENABLE();
3
4     DMA_InitStructure.Channel = DMA_CHANNEL_6;
5     DMA_InitStructure.Direction = DMA_MEMORY_TO_PERIPH;
6     DMA_InitStructure.MemInc = DMA_MINC_ENABLE;
7     DMA_InitStructure.PeriphInc = DMA_PINC_DISABLE;
8     DMA_InitStructure.PeriphDataAlignment = DMA_PDATAALIGN_BYTE;
9     DMA_InitStructure.MemDataAlignment = DMA_MDATAALIGN_BYTE;
10    DMA_InitStructure.Mode = DMA_NORMAL;
11    DMA_InitStructure.Priority = DMA_PRIORITY_VERY_HIGH;
12    DMA_InitStructure.FIFOMode = DMA_FIFOMODE_DISABLE;
13    DMA_InitStructure.MemBurst = DMA_MBURST_SINGLE;
14    DMA_InitStructure.PeriphBurst = DMA_PBURST_SINGLE;
15
16    DMA_Handle.Instance = DMA2_Stream5;
17    DMA_Handle.Init = DMA_InitStructure;
18
19    HAL_DMA_Init(&DMA_Handle);
20    HAL_DMA_Start(&DMA_Handle, (uint32_t) trame, (uint32_t) &GPIOA->ODR, LG_TRAME );
21 }
  
```

# Initialisation du timer et association avec le DMA

```
1 void InitTimer1() {
2     __TIM1_CLK_ENABLE();
3
4     TIM_TimeBaseStructure1.Prescaler = 5000;
5     TIM_TimeBaseStructure1.CounterMode = TIM_COUNTERMODE_UP;
6     TIM_TimeBaseStructure1.Period = 0x00FF;
7     TIM_TimeBaseStructure1.ClockDivision = TIM_CLOCKDIVISION_DIV1;
8     TIM_TimeBaseStructure1.RepetitionCounter = 0;
9
10    s_TimerInstance1.Init = TIM_TimeBaseStructure1;
11    s_TimerInstance1.Instance = TIM1;
12
13    HAL_TIM_Base_Init(&s_TimerInstance1);
14    TIM1->DIER = TIM_DMA_UPDATE;    // DMA Interrupt Enable
15    HAL_TIM_Base_Start(&s_TimerInstance1);
16 }
```

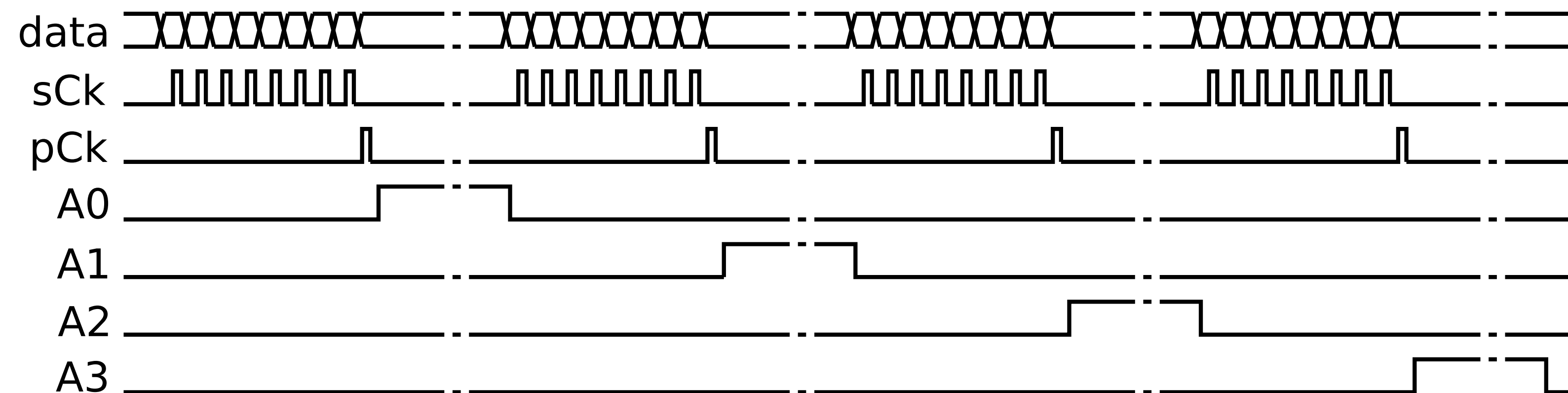
# Utilisation du DMA pour piloter des matrices de LED

---

- Les registres série-parallèles nécessitent des signaux de données et des horloges

# Utilisation du DMA pour piloter des matrices de LED

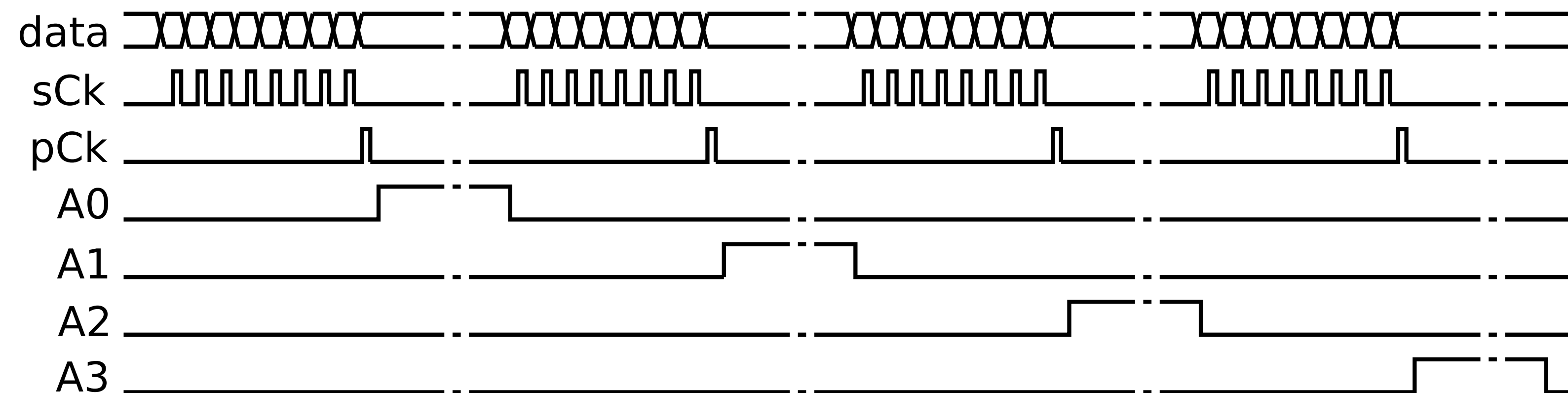
- Les registres série-parallèles nécessitent des signaux de données et des horloges





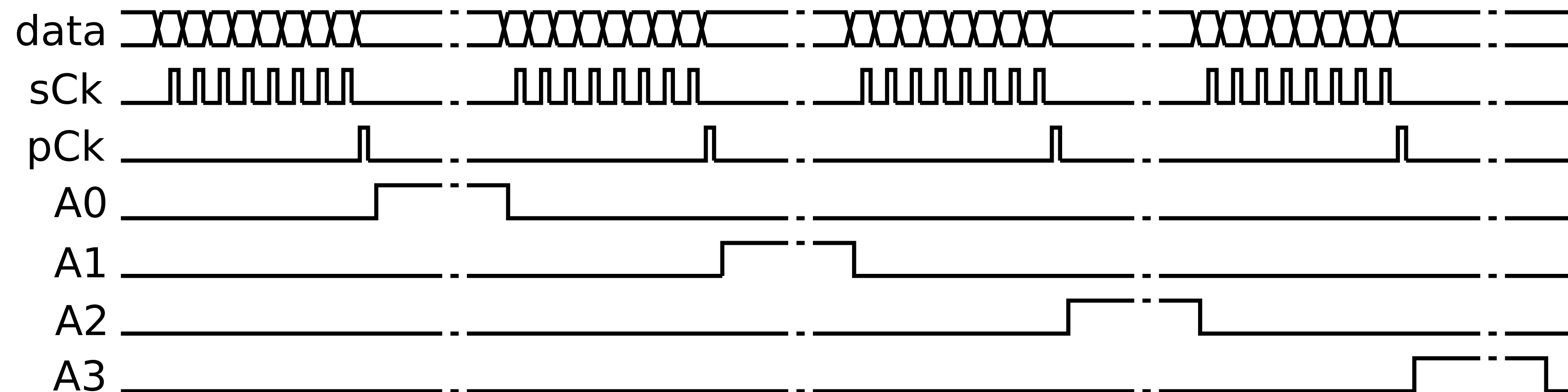
# Utilisation du DMA pour piloter des matrices de LED

- Les registres série-parallèles nécessitent des signaux de données et des horloges
- Les horloges peuvent être placées en mémoire comme les données



# Utilisation du DMA pour piloter des matrices de LED

- Les registres série-parallèles nécessitent des signaux de données et des horloges
- Les horloges peuvent être placées en mémoire comme les données
- Certains microcontrôleurs ont des circuits programmables capables de générer de telle horloges



# Accès direct en mémoire (DMA)

- Limite du débit d'un microcontrôleur
- DMA, économie des instructions de transfert
- Exemple simple... mais programme complexe
- Application à une matrice de LED
- Génération des horloges