

SYNTHÉTISEUR SID

Guide technique : Recréer la puce MOS 6581/8580 du Commodore 64

Le **SID (Sound Interface Device)** est sans doute la puce sonore la plus iconique de l'ère 8-bits. Conçue par Robert Yannes en 1981, elle a transformé le Commodore 64 en un véritable instrument de musique. Ce tutoriel explore comment recréer son architecture en synthèse numérique (DSP/FPGA/Logiciel).

1. Architecture Globale

Le SID possède 3 voix de synthèse indépendantes. Chaque voix est identique et comprend :

- Un oscillateur numérique (DCO).
- Un générateur d'enveloppe ADSR.
- Un modulateur (Ring Modulation / Hard Sync).

Le tout passe ensuite par un **filtre multimode analogique unique** (commun aux 3 voix).

2. L'Oscillateur (DCO)

Contrairement à un oscillateur analogique pur, le SID utilise un accumulateur de phase 24-bits. La fréquence de sortie est calculée comme suit :

$$f_{out} = (f_{clk} / 16777216) \times F_{reg}$$

Où f_{clk} est la fréquence d'horloge du système (~1 MHz) et F_{reg} la valeur du registre de fréquence (16 bits).

Formes d'onde

- **Triangle** : Générée à partir des bits de l'accumulateur.
- **Dent de scie** : Lecture directe des bits de poids fort de l'accumulateur.
- **Impulsion (PWM)** : Comparaison entre l'accumulateur et un registre de largeur d'impulsion.
- **Bruit** : Utilisation d'un registre à décalage à réaction linéaire (LFSR) 23-bits.

3. Le Filtre SID (Multimode)

C'est la partie la plus difficile à émuler car elle était analogique et imparfaite sur la puce originale. Le filtre est un filtre à état variable (State Variable Filter) 12dB/octave.

```
// Pseudo-code d'un filtre d'état variable (SVF)
low = low + f * band;
high = input - low - q * band;
band = f * high + band;
// Sortie selon le mode : lowpass, highpass ou bandpass
```

4. Implémentation Logicielle (C++)

Voici une structure de base pour simuler une voix SID simplifiée :

```
class SIDVoice {
    uint32_t accumulator;
    uint32_t frequency;
    float pulsewidth;

    float process() {
        accumulator = (accumulator + frequency) & 0xFFFFF;
        uint16_t msb = accumulator >> 8;

        // Exemple : Dent de scie
        float sample = (float)msb / 65536.0f;

        // Application de l'ADSR (simplifié)
        sample *= adsr.getNextAmplitude();

        return sample;
    }
};
```

```
}  
};
```

5. Les Spécificités "Culte"

Hard Sync & Ring Modulation

La **Hard Sync** force l'accumulateur d'une voix à se réinitialiser lorsque l'accumulateur d'une autre voix passe par zéro. Cela crée des harmoniques métalliques très riches.

La **Ring Mod** effectue un OU-Exclusif (XOR) entre les MSB des deux oscillateurs, produisant des sons de type "cloche".

6. Liste des Registres (Mémoire)

Adresse	Fonction	Description
\$D400 - \$D401	Freq Voice 1	Fréquence (LO/HI)
\$D404	Control Voice 1	Waveform, Gate, Sync, Ring
\$D405 - \$D406	ADSR Voice 1	Attack/Decay et Sustain/Release
\$D417	Filter Control	Resonance et routage des voix