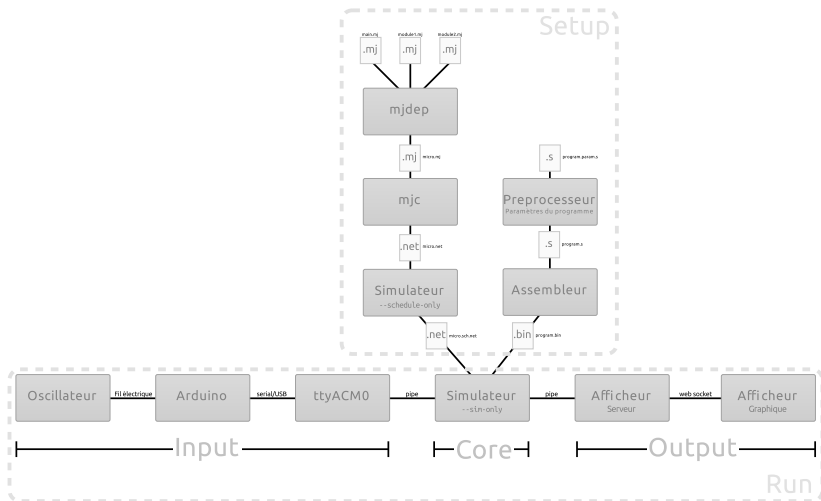


# Projet de microprocesseur

Nicolas J., Elie M., Aurélien D. et Louis G.

28 janvier 2014

# Vue d'ensemble



# Sommaire

Simulateur

Microprocesseur

Programme

Afficheurs

Oscillateur

# Simulateur - Fonctionnement général

- ▶ Parseur de netlist
- ▶ Optimizer (ordonne et simplifie les netlists)
- ▶ Parseur de fichier de ROM
- ▶ Lecture/Écriture des entrées/sorties du circuit
- ▶ Initialisation
- ▶ Cœur du simulateur

## Optimizer

- ▶ Successeur du Scheduler
- ▶ Suppression des nappes (sauf vers RAM et ROM)
- ▶ Suppression des doublons
- ▶ Problèmes de complexité

## Optimizer

- ▶ Successeur du Scheduler
- ▶ Suppression des nappes (sauf vers RAM et ROM)
- ▶ Suppression des doublons
- ▶ Problèmes de complexité

## Simulation

- ▶ Problèmes d'optimisation
- ▶ Phase d'initialisation pour modifier la structure des données

## ROM

- ▶ Contient le programme à exécuter.
- ▶ Décrit par un fichier .bin, suite de 0 et 1 pouvant contenir des commentaires.

## ROM

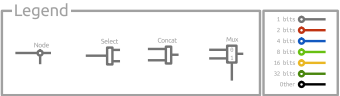
- ▶ Contient le programme à exécuter.
- ▶ Décrit par un fichier .bin, suite de 0 et 1 pouvant contenir des commentaires.

## Entrées/Sorties

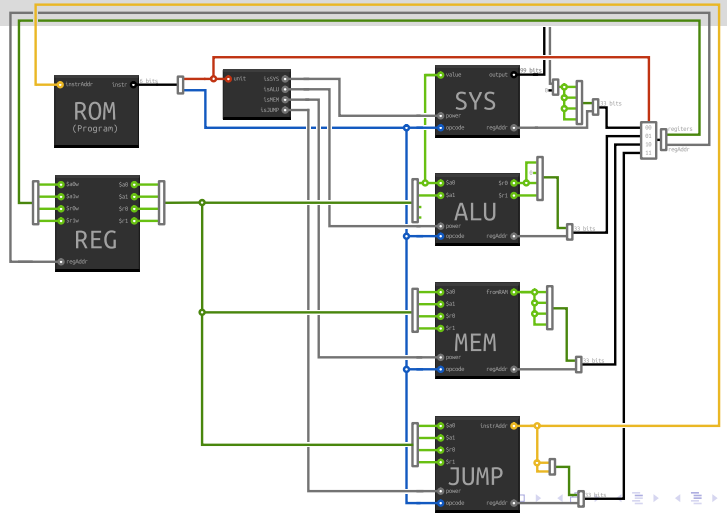
- ▶ Utilisations de l'entrée/sortie standard
- ▶ Communication avec les périphériques par pipe
- ▶ Possibilité d'asynchronisation de l'entrée



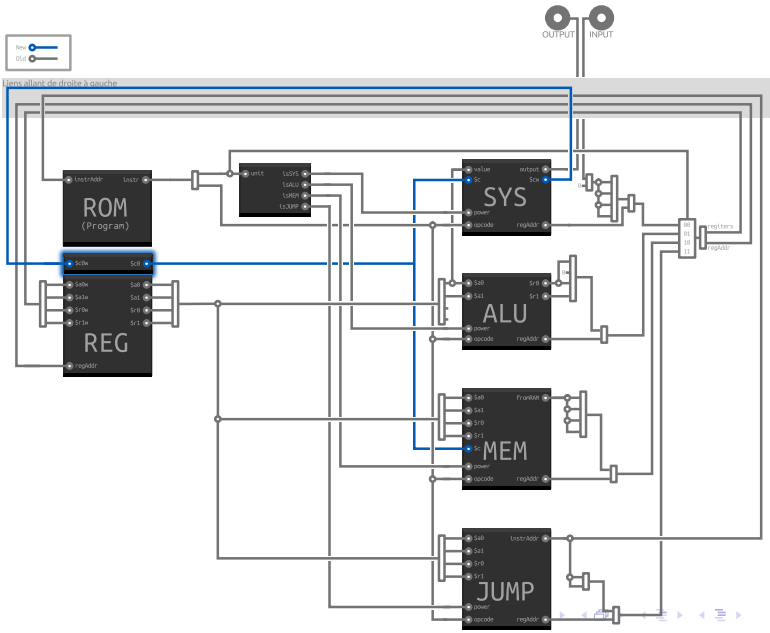
# Microprocesseur - Architecture



Liens allant de droite à gauche



# Microprocesseur - Modification

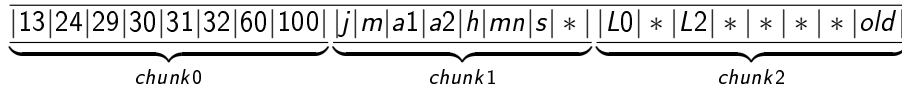


# Microprocesseur - Séparation CPU/GPU

# Programme

- ▶ Lexer "compilateur"
- ▶ Initialisation de la RAM
- ▶ LUT dans la RAM à l'initialisation
- ▶ Réinitialisation de la LUT au changement d'années (pour traiter les années bisextiles)
- ▶ Indentation pour connaître la valeur de c

## Programme - Gestion de la mémoire



## Programme - Difficultés

- ▶ Difficile de coder avec si peu d'instructions, mais intéressant
- ▶ Surtout lorsqu'il a fallu prendre en compte le nombre de jours par mois
- ▶ LUT dans la RAM à l'initialisation
- ▶ Réinitialisation de la LUT au changement d'années (pour traiter les années bisextiles)

# Programme - Compilation

- ▶ Langage plus haut niveau pour créer le code
- ▶ Pile et données statiques
- ▶ Valeurs des Li6 chargés au début
- ▶ Vérification : valeur de c
- ▶ Optimisation difficile

# Afficheur non graphique

## Afficheur heure et date

- ▶ Lecture en continue
- ▶ Décodage 7 segments → chiffre



# Afficheur non graphique

## Afficheur heure et date

- ▶ Lecture en continue
- ▶ Décodage 7 segments → chiffre

## Afficheur stats

- ▶ Secondes « réelles »
- ▶ Fréquences (moyenne et instantanée)

# Afficheur graphique

## Serveur

- ▶ Serveur NodeJS
- ▶ Même base que l'afficheur non graphique
- ▶ Permet de communiquer avec le client en temps réel
- ▶ Envoi toutes les 100ms

# Afficheur graphique

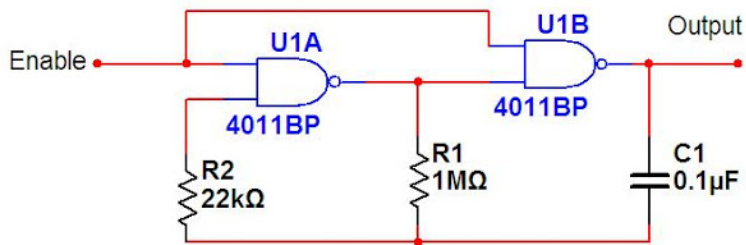
## Serveur

- ▶ Serveur NodeJS
- ▶ Même base que l'afficheur non graphique
- ▶ Permet de communiquer avec le client en temps réel
- ▶ Envoi toutes les 100ms

## Client

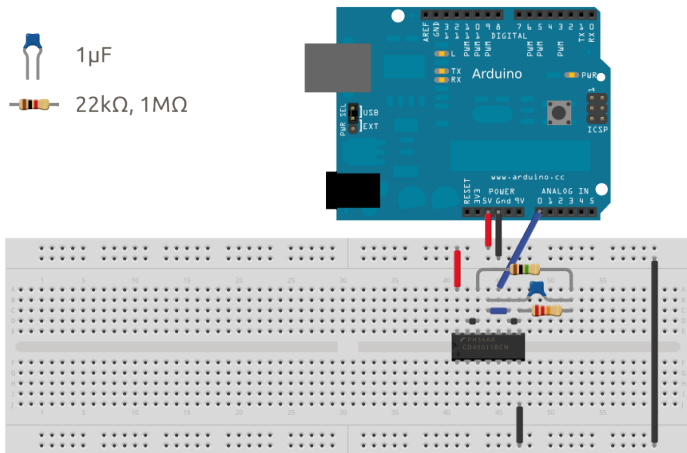
- ▶ Inspiré d'un exemple de la bibliothèque graphique RaphaëlJS
- ▶ Ne gère pas correctement le nombre de jours par mois

# Oscillateur



[easy-electronics4u.blogspot.com](http://easy-electronics4u.blogspot.com)

# Oscillateur - Montage



# Oscillateur - Mesures pratiques

