

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN
FACULTE DU GENIE ELECTRIQUE
DEPARTEMENT D'ELECTRONIQUE
LICENCE ELECTRONIQUE : L3
MODULE: mini-projet



POLYCOPIE DES TRAVAUX PRATIQUES:
Introduction à la simulation et routage des circuits avec le logiciel
PROTEUS V7 et V8

PREPARE PAR:
M. LAKHDARI.F

USTO-MB 2016/2017

SOMMAIRE

1. Initiation à Isis Proteus	1
1.1 L'environnement de travail de Proteus	1
1.2 La barre d'outils principale	2
2. L'analyse des circuits RLC de base	5
2.1 Le circuit RC et CR	5
2.2 Le circuit RL et LR	5
2.3 Le circuit RLC	5
3. Les Amplificateurs Opérationnels	6
3.1 Montages de bases	6
3.2 Filtre actif	7
4. Le Circuit Multifonctions (NE555)	8
4.1 Présentation du circuit	8
4.2 Principe et diagramme fonctionnel	8
4.3 Montage en monostable	9
4.4 Montage en astable de base	10
4.5 Montage en astable (MLI)	10
5. La simulation des Circuits Logiques	12
5.1 L'Analyseur logique	12
5.2 Les portes logiques TTL et CMOS	13
5.3 La bascule S-R Asynchrone	14
5.4 La bascule S-R Synchrone (SRH)	14
5.5 Encodeur et Afficheur 7 Segments	15
6. La simulation des microcontrôleurs	16
6.1 Simulation et configuration de base	16
6.2 Le mode pas à pas	17
6.3 Rappel sur MPLAB	19
6.4 L'Association de PROTEUS à MPLAB	21
7. Le module ARES de conception des circuits imprimés	22
7.1 Présentation de l'interface ARES	22
7.2 Routage automatique	22
Annexe	25

Introduction

Ce document est destiné en particulier aux étudiants de la troisième année licence (option: électronique), dont une partie du module 'mini-projet' consiste à étudier et à maîtriser un logiciel de simulation des circuits et systèmes électroniques.

Il s'agit d'une initiation au logiciel de simulation ISIS-PROTEUS, qui est largement utilisé ces dernières années comme outil pédagogique, en particulier pour sa simplicité due à son environnement graphique et interactif.

Le document est rédigé d'une façon relativement simple, avec un niveau de difficulté progressif, afin qu'il soit même utile aux étudiants des autres options, et même des autres niveaux

La finalité est de doter l'étudiant d'un outil, lui permettant en premier lieu une révision et une assimilation de quelques notions de base d'électricité et d'électronique. Cet outil facilitera aussi, par la suite la tâche de l'étudiant durant la réalisation de son mini-projet.

Les différents outils mis à la disposition de l'utilisateur dans ISIS-PROTEUS, permettent, outre la simulation, des analyses dans des domaines différents (temporel, fréquentiel), ainsi que des régimes différents (statique, dynamique).

Les différents outils du logiciel sont décrits au fur et à mesure, via une série d'exemples et d'applications, couvrant différents domaines de l'électronique, allant des circuits analogiques simples (RLC, circuits à base de diodes et transistors, AOP,..) aux circuits numériques y compris les microcontrôleurs.

Certaines applications sont validées par des tests expérimentaux effectués au niveau du laboratoire, afin de doter l'étudiant d'un esprit critique lui permettant d'analyser et de comparer les résultats obtenus, théoriquement, expérimentalement, et par simulation informatique.

Le module ARES du logiciel PROTEUS, dédié à la conception des circuits imprimés est aussi présenté en bref vers la fin du document, en gardant la même approche basée sur une série d'exemple et d'applications.

Initiation à Isis PROTEUS

Isis Proteus est un logiciel de développement et de simulation d'application via un environnement graphique simple et interactif.

Ce TP est une initiation à ce logiciel, basé sur un ensemble d'exemples, dont les résultats doivent être comparés par la suite à ceux obtenus théoriquement et expérimentalement.

I- 1. L'environnement de travail de PROTEUS

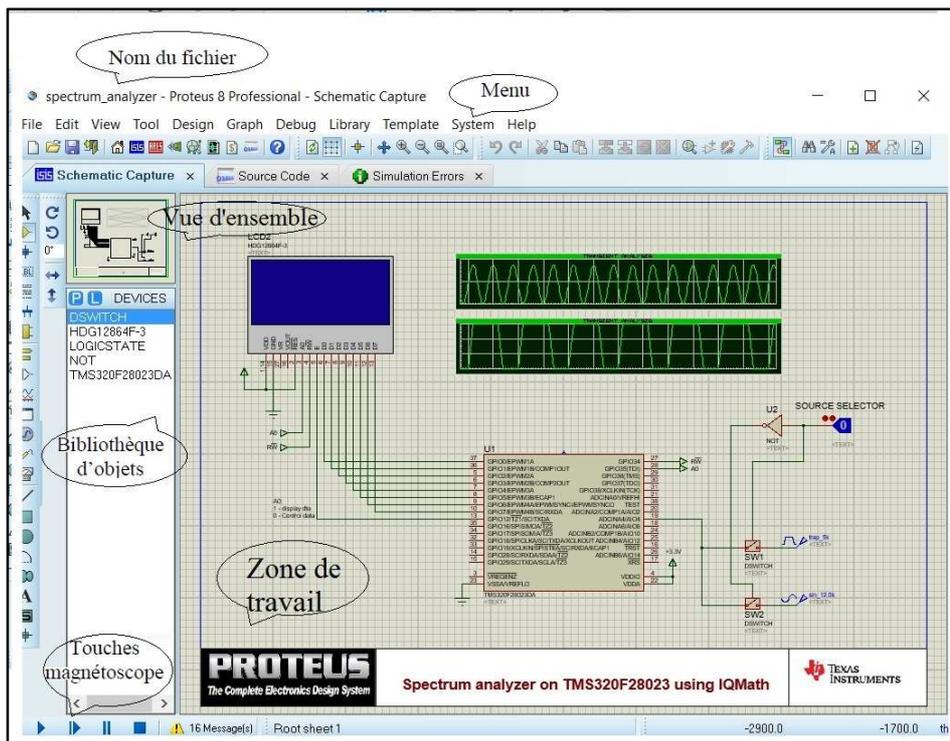
Le lancement de PROTEUS donne un environnement classique de type Windows, constitué d'une fenêtre principale, et d'un ensemble de barres d'outils .

Outre le menu classique permettant la gestion des fichiers, de l'affichage, et des options des projets,

la fenêtre principale comprend **une Zone de travail** destinée au développement des circuits à simuler et à tester.

Une Bibliothèque d'objets affiche la liste des objets (circuits électriques, électroniques,...) utilisés dans l' application en cours.

les différentes **Touches magnétoscope** constituées des raccourcis permettant le lancement de la simulation, ainsi que la mise en pause, l'exécution pas à pas, et l'arrêt de la simulation.



1.2 La barre d'outils principale

La boîte verticale de boutons, comprend les principaux raccourcis nécessaires au développement rapide d'applications.

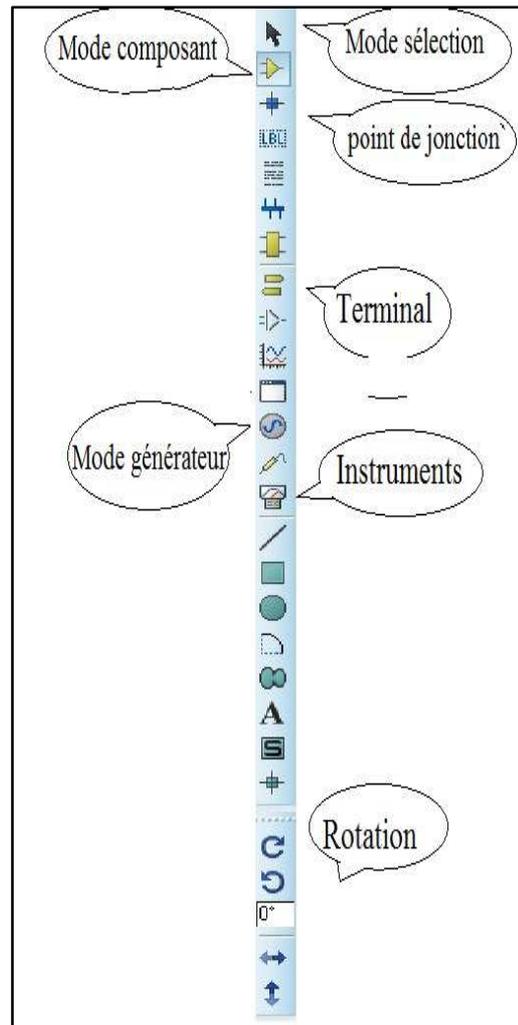
En plaçant le curseur de la souris sur un bouton, sans cliquer, une info-bulle affiche le nom du bouton, dont les plus importants sont:

- **Component mode (Mode Composant)**,
Un clic sur ce bouton puis *Component from libraries*), permet l'ajout de différents composants

- **Generator Mode (Mode Générateur)**
permettant l'accès aux différents types de générateurs

- **Instruments**
un raccourci permettant l'ajout des appareils de mesure tels que le voltmètre, l'ampèremètre, et l'oscilloscope.

- **Terminal**
permettant d'ajouter des points particuliers dans un schéma tels que les entrées/sorties ou Gnd



Application 1

- Lancer Proteus, puis éditer et simuler le circuit ci-contre.
- Justifier les valeurs numériques obtenues.

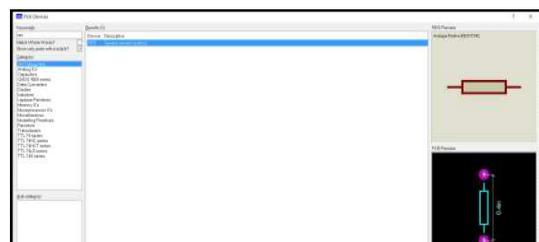
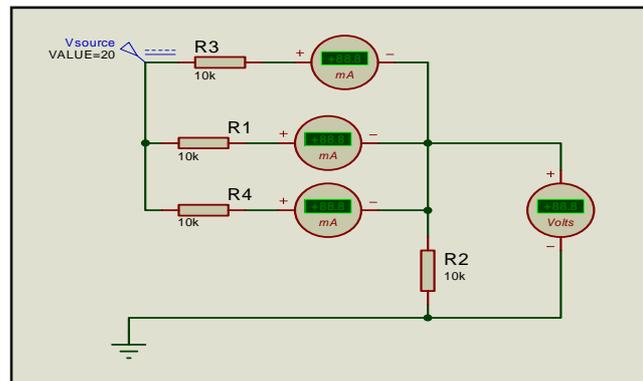
Remarque 1:

les composants utilisées sont :

- source de tension continue (Mode générateur) de 20v.
- des résistances (modèle RES) de 10k
- un voltmètre, et des ampèremètres (mode Instruments)
- La référence GND (mode Terminal)

Remarque 2:

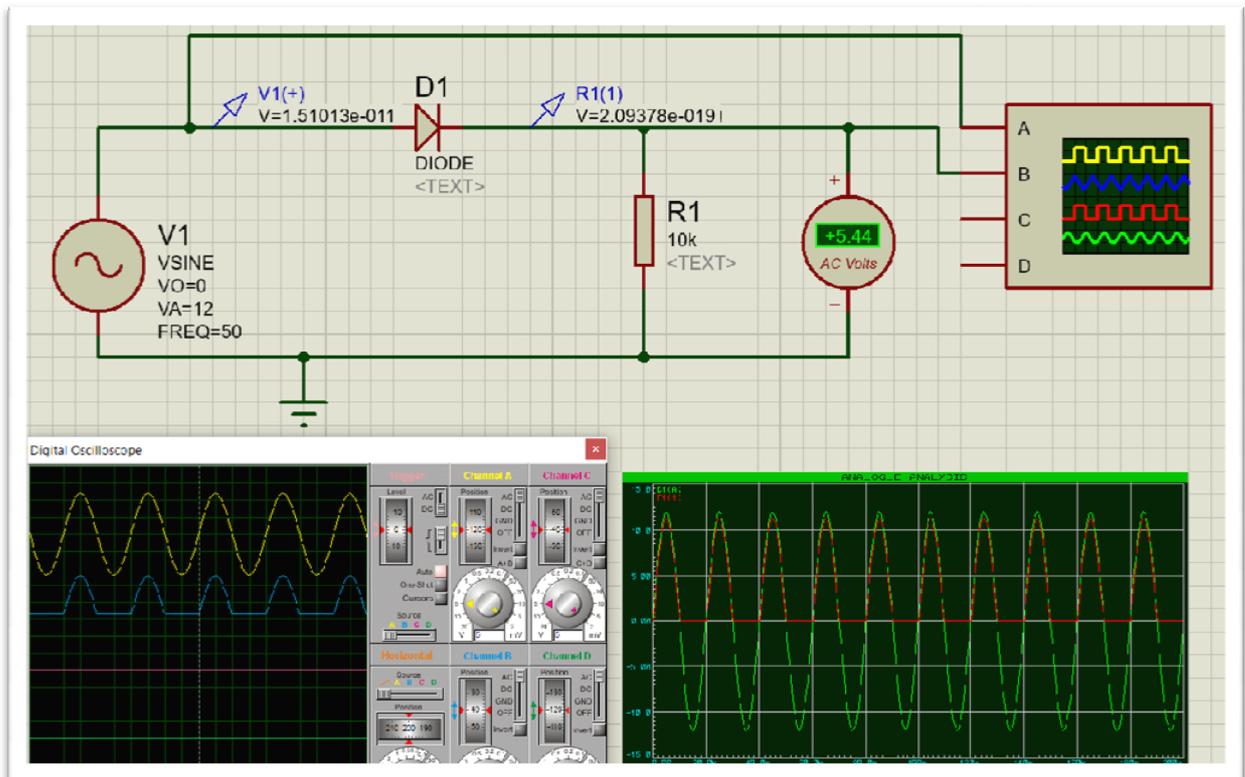
les composants (par exemple RES) peuvent être obtenus soit à partir de leur bibliothèque, soit via une recherche directe.



Application 2

Dans cette application on s'intéresse au redressement d'un signal alternatif par diode.
Outre le voltmètre et la charge résistive, on utilisera:

- Une diode (modele générique) de la bibliothèque "Diode"
- Une source alternative Vsine de la bibliothèque "Simulaor Primitives"
- Un oscilloscope (mode Instruments)
- Une fenêtre analogique (mode Graph)
- Deux sondes de tension (mode Probe)



- Simuler le schéma proposé
- Modifier le schéma pour avoir un redressement double alternance.
- Comparer le taux d'ondulation théorique des deux signaux obtenus
- Ajouter en sortie, un condensateur de filtrage (Modèle CAP) de 470UF, et visualiser la sortie.
- Ajouter en sortie deux régulateurs 7805, 7812 et visualiser la sortie

Remarque 3:

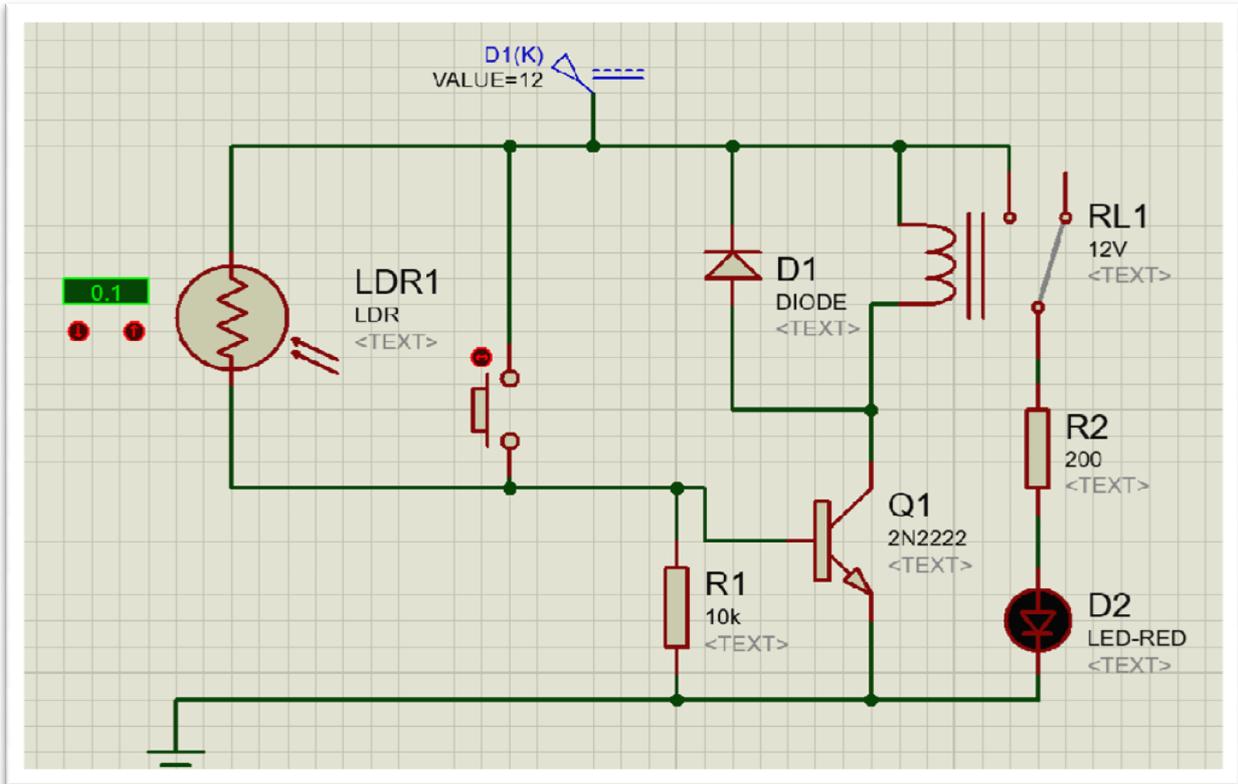
- Pour affecter une sonde à un graphe, il est nécessaire de sélectionner la sonde représentative du signal à visualiser. Puis glisser la sonde dans le graphe.
- Pour lancer la simulation, le graphe doit être Pointé tout en appuyant sur la **touche 'Espace'**
- Un double clic sur le graphe permet de fixer le temps de simulation à afficher (par exemple 200ms pour cet exemple).

Remarque 4:

Si l'oscilloscope ne s'affiche pas, Cliquer sur "debug / Oscilloscope"

Application 3

Le circuit suivant illustre un exemple de la commande d'un Relais (Relay), en utilisant un bouton poussoir (Button) et une LDR



- Définissez le principe d'une LDR
- Définissez le principe d'une LED
- Définissez le principe de fonctionnement d'un relais et de sa commande.
- Simuler et vérifier le fonctionnement du circuit.
- Dans le schéma proposé, justifier l'utilité de la diode.

L'analyse des circuits RLC de base

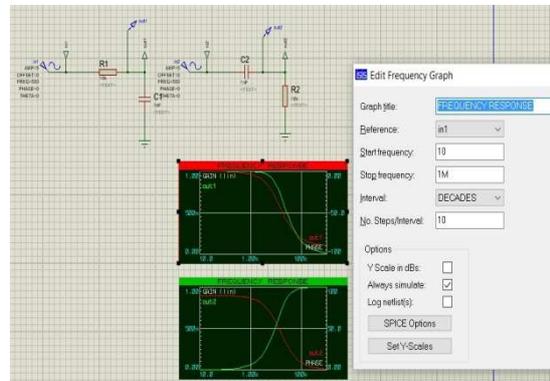
Dans cette partie on s'intéresse à l'analyse des circuits RLC de base, l'objectif est de maîtriser certains outils de PROTEUS permettant en particulier l'analyse fréquentielle et la synthèse des fonctions de transfert des systèmes.

2.1 Le circuit RC et CR

Sous PROTEUS, il est possible de déterminer la fonction de transfert d'un système en exploitant le graphe "Frequency response", et en précisant l'entrée et la sortie du système.

Application 1:

- Lancer PROTEUS, et éditer le schéma ci-contre comprenant deux circuits un RC, et un CR.
- Chaque circuit comprend une pin d'entrée (in1,in2) et une pin de sortie (out1,out2). Les deux pins sont issue du mode terminal.
- Pour obtenir la réponse en fréquence de chaque circuit un graphe de type 'frequency response' est ajouté, en modifiant sa référence soit à 'in1' dans le premier cas soit à 'in2' dans le deuxième cas.



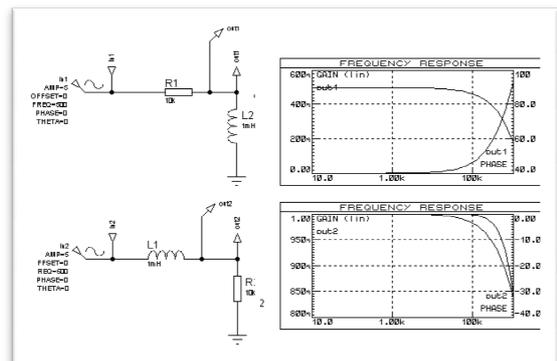
remarque :

Une sonde de tension doit être ajoutée, puis glissée une fois vers la droite, et une deuxième fois vers la gauche du graphe, cela permettra d'obtenir respectivement la variation de l'amplitude et de la phase du circuit en fonction de la fréquence.

- Comparer les résultats de la simulation avec ceux obtenus théoriquement.

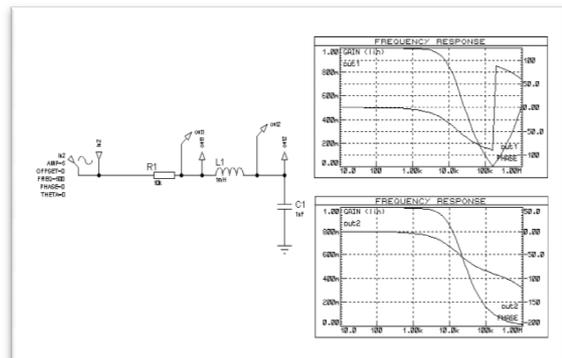
2.2 Le circuit RL et LR

Simuler les deux circuits RL et LR du schéma ci-contre, et comparer les réponses fréquentielles obtenues avec ceux déterminées théoriquement.



2.3 Le circuit RLC

Simuler et analyser théoriquement les réponses fréquentielles du circuit RLC donnée dans le schéma suivant.



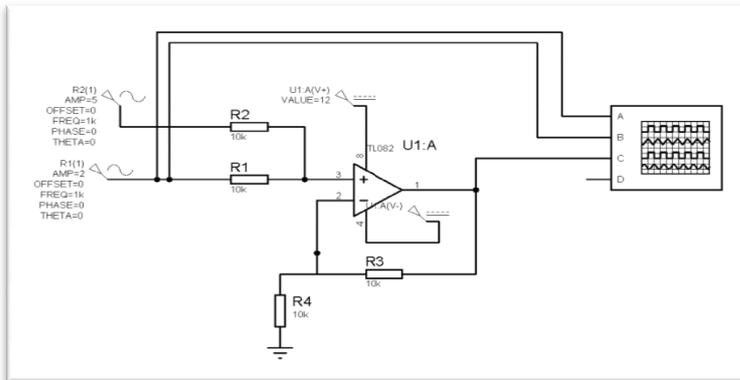
Les Amplificateurs Opérationnels

Dans cette partie on s'intéresse à des montages à base d'amplificateurs opérationnels. On utilisera pour cette raison le circuit TL082 (voir le datasheet donné en annexe).

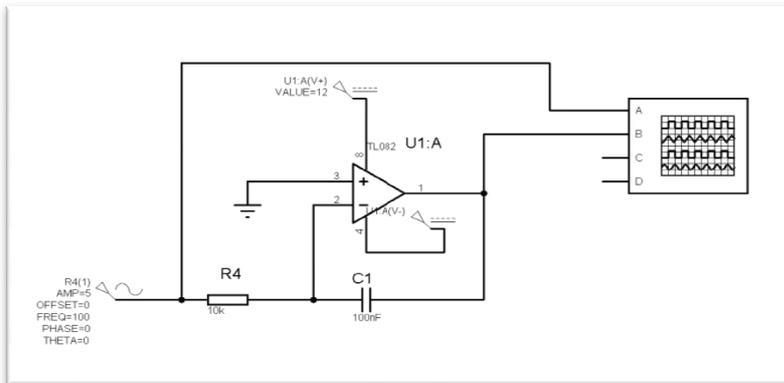
3.1. Montages de bases

Pour chaque montage il est nécessaire de déterminer théoriquement la fonction réalisée et de simuler le fonctionnement.

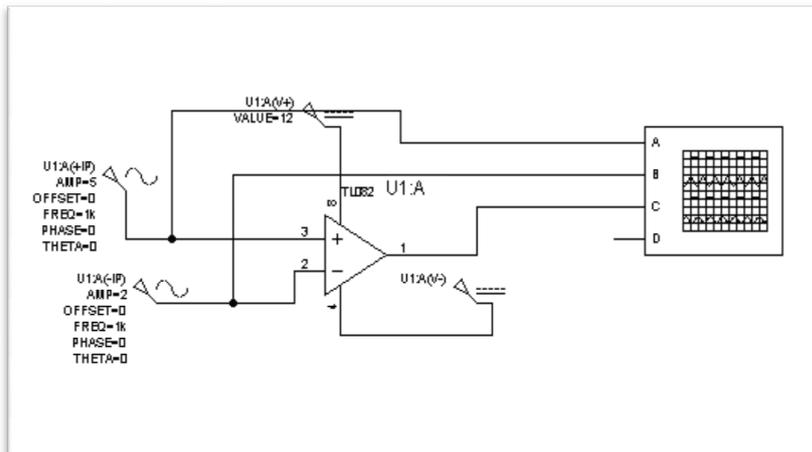
Montage 1 :



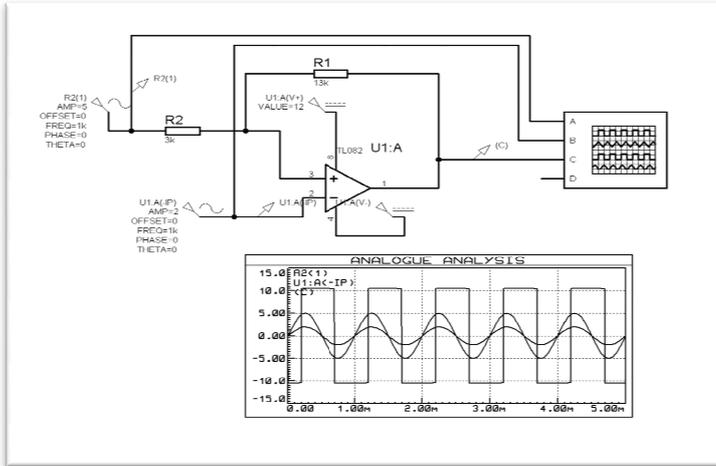
Montage 2 :



Montage 3 :



Montage 4:

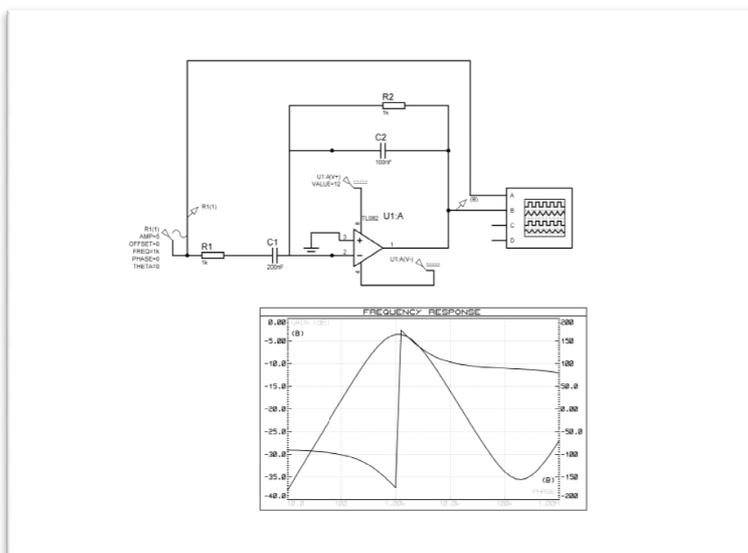


3.2 Filtre actif

- En considérant le montage suivant, déterminer théoriquement l'expression de sa réponse fréquentielle.

- Comparer le résultat avec celui obtenu en utilisant le module 'Frequency response' de PROTEUS

-Réaliser expérimentalement le montage et prélever point par point la réponse fréquentielle du circuit, et comparer avec les résultats de la simulation.



Le Circuit Multifonctions (NE555)

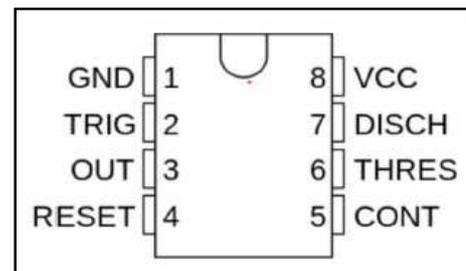
Cette partie est consacrée à la simulation et à la réalisation de montages élémentaires à base du circuit intégré NE555. Ce dernier a la particularité de permettre la réalisation de différentes fonctions électroniques liées à la gestion de temporisation.

4.1 Présentation du circuit

Le NE555 est un circuit utilisé en particulier pour la gestion des temporisation en mode multivibrateur. ce circuit a été créé et commercialisé au début des années soixante-dix, mais en raison de sa simplicité, de son faible cout, et de sa stabilité, il est toujours utilisé dans les circuits 'timer' de base.

le circuit est disponible généralement en boîtier DIL (dual in line) de 8 pates, dont les appellations et les fonctions sont résumées dans le tableau suivant:

N°	Nom	Description
1	GND	La masse
2	TRIG	Pin D'amorçage de la temporisation (détection d'une tension inférieure à 1/3 VCC
3	OUT	Signal de sortie
4	RESET	Réinitialisation /Interruption de la temporisation
5	VCC	L'alimentation 5v à 18v
6	DISCH	Pin permettant la décharge du condensateur de la temporisation
7	THRES	Pin déclenchant la fin de temporisation (la tension atteint en montant 2/3 VCC)
8	CONT	Pin donnant accès à la référence interne (2/3 VCC)

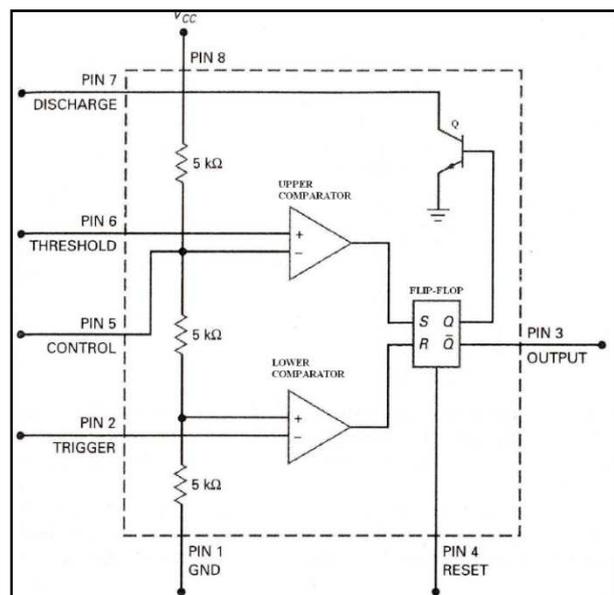


4.2 Principe et diagramme fonctionnel

Le NE555 est constitué de 23 transistors, 2 diodes, et 16 résistances.

Du point de vue fonctionnel, le circuit peut être décrit par le schéma ci-contre, comprenant :

- Deux circuit comparateurs
 - Un diviseur de tension de 3 résistances identiques de 5k.
- Les tensions de référence sont respectivement 1/3 et 2/3 VCC, et servent comme référence aux deux comparateurs
- Une bascule SR (avec pin reset) contrôlée en entrée par les deux comparateurs
 - Un transistor permettant la décharge du condensateur de temporisation.



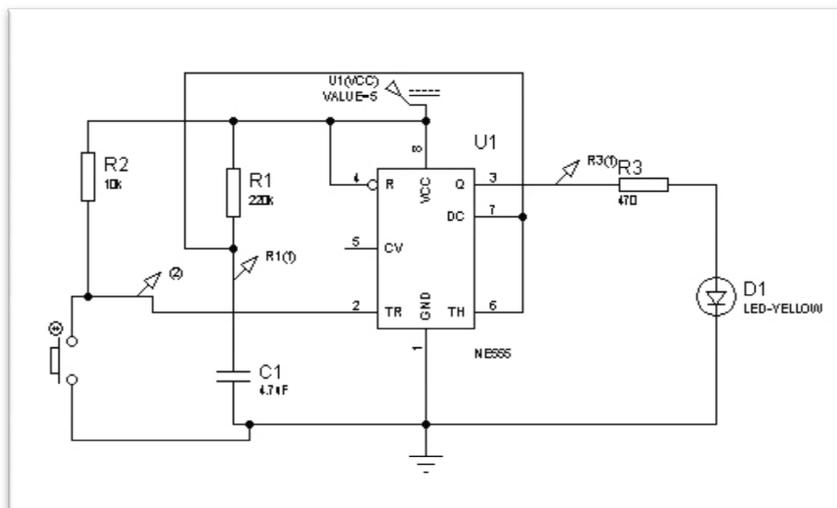
le fonctionnement du circuit peut être résumé comme suit:

- Si **RESET=0** : La bascule est remise à zéro, le transistor de décharge s'active et la sortie reste à un niveau bas.
- Si **TRIGGER < VCC/3** : la bascule est activée (SET) et la sortie est à un niveau haut, le transistor de décharge est désactivé.
- Si **THRESHOLD > VCC.2/3** : la bascule est remise à zéro (RESET) et la sortie est à un niveau bas, le transistor de décharge s'active.
- Dans le cas ou **THRESHOLD < VCC.2/3** et **TRIGGER > VCC/3** , alors la bascule conserve son état précédent, ainsi que la sortie et le transistor de décharge.

4.3 Montage en monostable

Le Monostable est un circuit possédant un seul état stable, qu'il quitte uniquement après une impulsion sur son entrée (Trigger).

Après une durée prédéfinie dans le second état, le circuit revient à son état initial.



- Simuler et vérifier le fonctionnement du circuit, en modifiant les valeurs de R1 et C1.
- Remplacer la partie déclenchement par un générateur d'impulsion, et visualiser dans un graph les trois signaux indiqués dans le schéma.
- Justifier théoriquement et expérimentalement les temporisations obtenues

4.4 Montage en astable de base

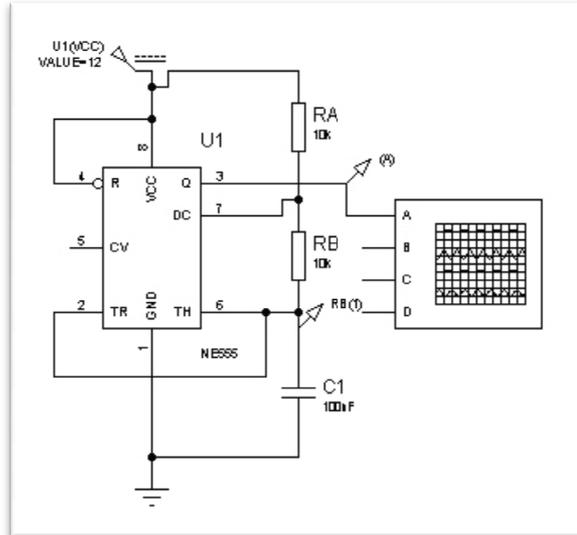
Dans le cas du montage astable, le circuit ne possède pas d'état stable, et par conséquent fonctionne comme oscillateur.

la fréquence d'oscillation et le rapport cyclique sont fixés avec deux résistances et un condensateur.

Une oscillation complète est effectuée lorsque le condensateur se charge de $1/3 V_{cc}$ jusqu'à $2/3 V_{cc}$.

Remarque

Durant la charge, les résistances RA et RB sont en série avec le condensateur, mais la décharge s'effectue uniquement à travers RB. Par conséquent le rapport cyclique peut être modifié.



- Simuler et vérifier le fonctionnement du circuit, en modifiant les valeurs de RA et C1.
- Visualiser dans un Graph les deux signaux indiqués dans le schéma.
- Déterminer les limites du rapport cyclique qu'on peut avoir avec ce schéma.
- Déterminer théoriquement l'expression de la fréquence, et du rapport cyclique du signal généré par ce montage
- Vérifier expérimentalement le fonctionnement.

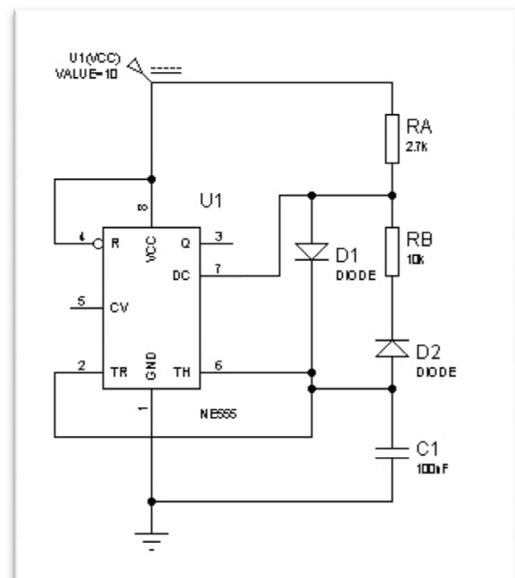
4.5 Montage en astable (MLI)

Afin de permettre une variation du rapport cyclique, c'est à dire une modulation de largeur d'impulsion (MLI) complète, le montage précédent est modifié en ajoutant une diode D1.

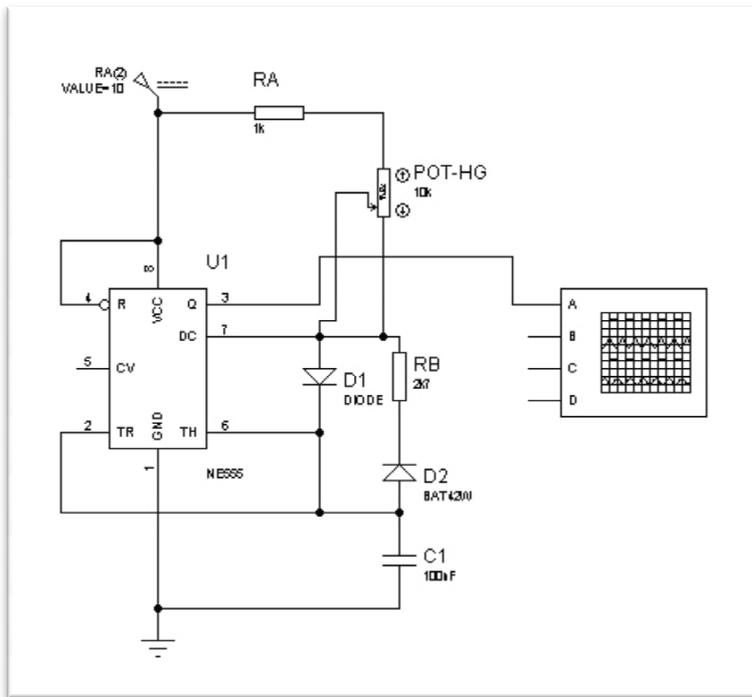
Due a cette diode, le condensateur se charge seulement via RA, vue que RB est court-circuitée durant cette phase. en contre partie la décharge s'effectue via RB

Remarque

La diode D2 montée en inverse dans le schéma est optionnelle, est permet un meilleur court-circuit de RB



Le montage suivant est souvent utilisé dans certaines applications, qui nécessitent une variation du rapport cyclique (sur l'intervalle 0-100%) via un potentiomètre



- Simuler et vérifier le fonctionnement du circuit, en modifiant les valeurs du potentiomètre.
- mesurer la fréquence du signal généré, ainsi que la limite basse et haute du rapport cyclique.
- Justifier théoriquement et vérifier expérimentalement les résultats obtenues

La simulation des Circuits Logiques

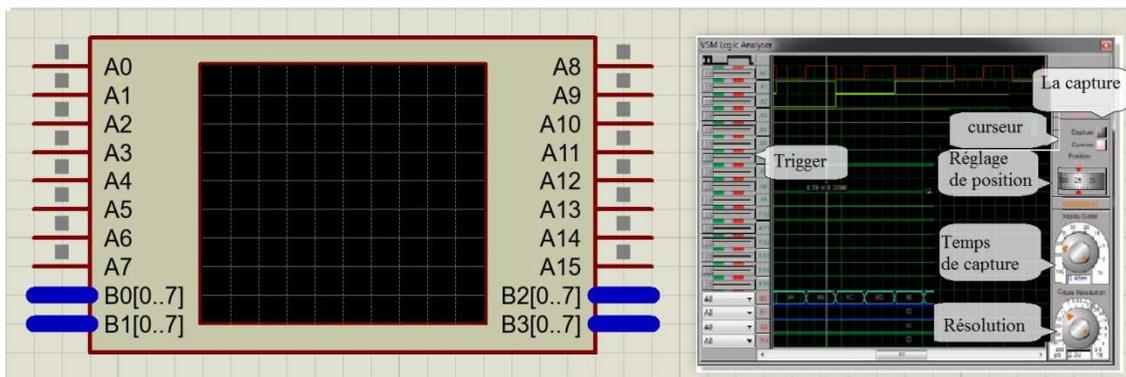
Dans cette partie on s'intéresse à la simulation par Proteus des circuits logiques de base. En plus des outils de d'analyse et de visualisation déjà décrits dans les sections précédentes, Proteus permet d'utiliser des pins d'entrée/sortie spécifiques aux signaux logiques, facilitant ainsi la tâche de simulation.

5.1 L'Analyseur logique

Cette instrument permet l'enregistrement continu de données numériques, via un processus d'échantillonnage dont la résolution définit l'impulsion minimale détectable.

La mémoire tampon de l'analyseur permet la sauvegarde de 10.000 échantillons, avec un control du temps de déclenchement de la capture des signaux.

Des mesures précises de largeur d'impulsion par exemple, sont possible grâce à un ensemble de marqueurs de mesure.

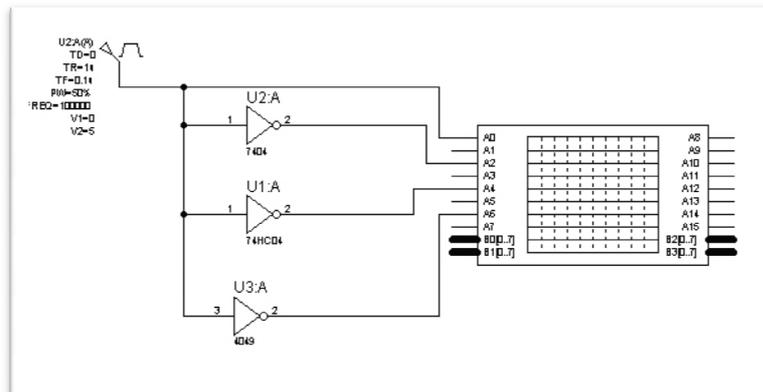


Les caractéristiques de l'analyseur (dépend de la version utilisée) sont :

- 16 Courbes 8x1 bit
- 4 Bus 2x8 bits
- Résolution de 0.5ns à 200us
- Temps de capture de 5ms à 2s
- Affichage réglable de 1 à 1000 échantillons/Division
- Deux curseur pour mesure de temps
- Déclenchement (Trigger) sur combinaison 'et' des état des entrées 'et/ou' fronts ainsi que les valeurs des bus.
- Réglage du déclenchement à un point entre 0 et 100% de la trame.

5.2 Les portes logiques TTL et CMOS

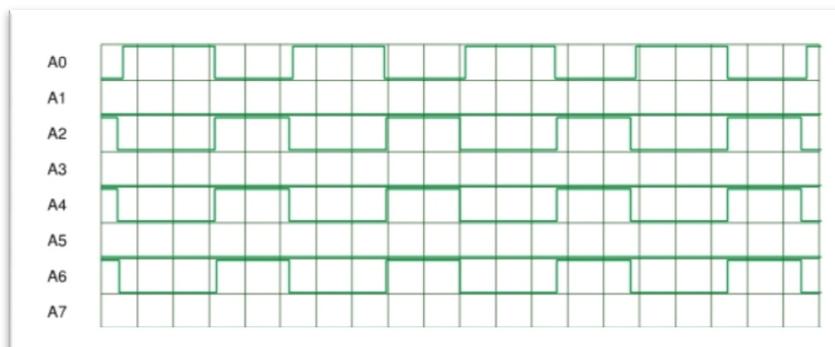
Dans cette application, on s'intéresse à la comparaison des performances des portes logiques (inverseurs) réalisées par des technologies de conceptions différentes (7404, 74HC04, CD4049).



- Réaliser, et simuler le schéma proposé
- Après le lancement de la simulation interactive procéder à un réglage adéquat de l'analyseur logique.

- * Le bouton résolution définit la petite largeur d'impulsion pouvant être enregistrée
- * Les conditions de déclenchement (Trigger) sont précisées dans la liste à droite du menu.
- * Le déclenchement de l'enregistrement des données est réalisé via le bouton Trigger, qui change de couleur (vert) durant le temps d'échantillonnage.

- En changeant le calibre d'affichage et le curseur de mesure, comparer les performances temporelles (delay time, rise time,...) des trois circuits logiques.

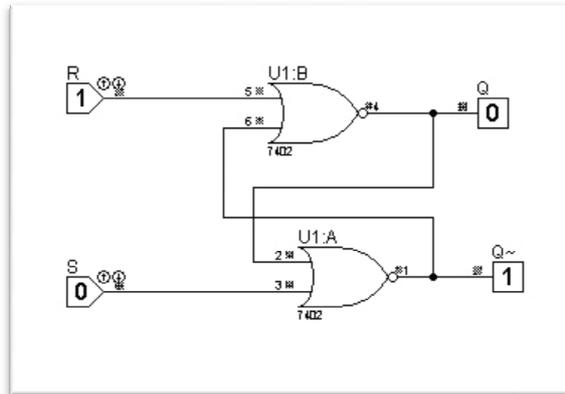


- Via une recherche bibliographique, comparer la technologie de conception de chaque circuit.

5.3 La bascule S-R Asynchrone

La bascule SR est un circuit séquentiel de base, et les autres types de bascules (synchrone ou asynchrone) se basent pratiquement sur cette bascule.

les différentes combinaisons des deux signaux d'entrée R et S permettent soit la mise à 1, soit la remise à 0, ou une mémorisation de l'état de la bascule. Pratiquement la réalisation de cette bascule peut prendre différentes formes, dont l'une est traitée par la suite.



La figure ci-dessus, illustre un exemple de réalisation d'une bascule SR avec des porte NOR (7402).

- simuler le circuit, et vérifier sa table de vérité.

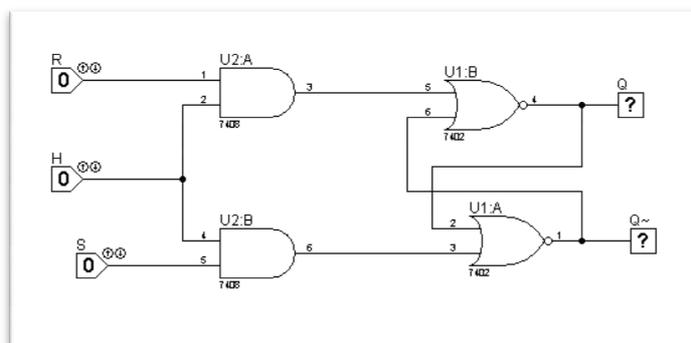
Remarque

Dans le schéma proposé sont utilisés des entrées analogiques (Logicstate) et des sondes logiques (Logicprobe)

5.4 La bascule S-R Synchrone (SRH)

Dans ce cas un signal d'horloge H synchronise le fonctionnement du circuit.

la bascule fonctionne comme une mémoire durant le niveau bas de H, et comme une bascule SR classique durant le niveau haut.

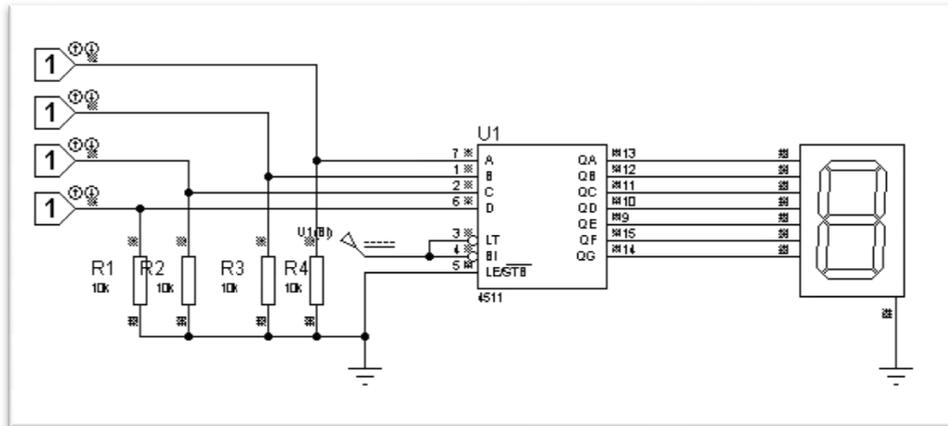


- Simuler le circuit proposé, et vérifier son fonctionnement et sa table de vérité.
- Modifier le schéma afin de permettre une visualisation des entrées/sorties sur un analyseur logique.

Pour cette finalité utiliser en entrée (S,R,H) des générateur de type **PWLin**.

5.5 Encodeur et Afficheur 7 Segments

Dans l'exemple suivants un décodeur BCD/7Segmen (4511) est utilisé pour un affichage sur un 7 segments à cathode commune.



- Réaliser le montage, puis prélever et vérifier la table de vérité.
- Modifier le montage en remplaçant le décodeur par un circuit à base de portes logiques élémentaires.

La simulation des microcontrôleurs

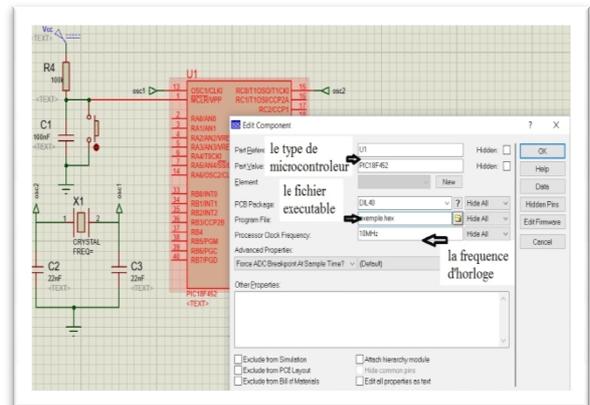
Dans cette partie on s'intéresse à la simulations des systèmes à base de Microprocesseurs / microcontrôleurs avec le logiciel Proteus.

L'initiation est faite via un ensemble d'applications; dont la totalité sont des exemples à base du microcontrôleur PIC18F452 de MICROCHIP, vu que' ce type de microcontrôleurs est déjà étudiés dans d'autres modules

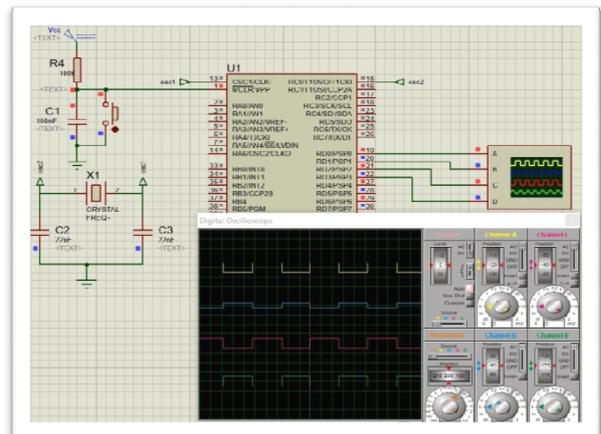
6.1 Simulation et configuration de base:

Le circuit de base d'un système à microcontrôleurs comprend généralement et en particulier (figure ci-contre):

- Un circuit RESET, qui peut être un simple circuit RC
- Un circuit d'horloge, qui est généralement un oscillateur à quartz.



Un double clic sur le microcontrôleur utilisé permet la définition de sa fréquence de fonctionnement, et du nom du fichier contenant le code machine (fichier.hex) qui sera exécuté durant la simulation.



Application 1

- Lancer Proteus, et développer le circuit à base du 18F452 ci-contre.
- Lancer MPLAB, puis éditer et compiler le programme assembleur suivant.
- Simuler le circuit précédent, en lui associant le programme donné.
- Déterminer la nature des signaux générées sur le PORTD.
- Modifier le programme pour réduire la fréquence des signaux générés au quart.

```

LIST P=18F452      ;directive
#include <P18F452.INC>
CONFIG OSC = HS, WDT=OFF

    ORG 0x0000
    CLRF PORTD
    CLRF TRISD
RET   MOVLW B'10101010'
      MOVWF PORTD
      CALL TEMP
      NOP
      NOP
      MOVLW B'01010101'
      MOVWF PORTD
      CALL TEMP
      GOTC RET

TEMP  MOVLW 0xA0
      MOVWF 0x20
REP   DECFSZ 0x20
      GOTC REP
      RETURN
      END
    
```

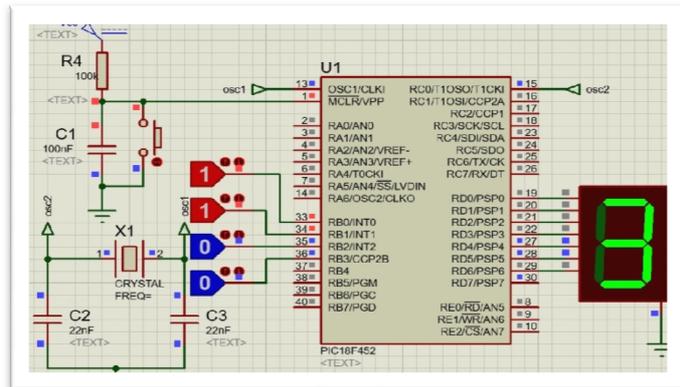
Application 2

- Lancer Proteus, et développer le circuit ci-contre, ou le port D du 18F452 est connecté a un 7 segment à cathode communes, les pins RB0-RB3 sont contrôlées par des entrées logiques.

- Lancer MPLAB, puis éditer et compiler le programme assembleur donné.

- Simuler le circuit précédent, en lui associant le programme donné, et vérifier le fonctionnement.

- Modifier le programme proposé, pour permettre un décodage complet BCD/7Segmen des 4 pins du PORTB ainsi qu'un affichage simultané sur le 7 segments



```

LIST P=18F452 ;directive
#include <P18F452.INC>
CONFIG OSC = HS, WDT=OFF

ORG 0x0000
CLRF PORTD
CLRF TRISD
SETF TRISB

RET    MOVLW 0x00
      CPFSEQ PORTB
      GOTC S1
      MOVLW 0x00
      MOVWF PORTD

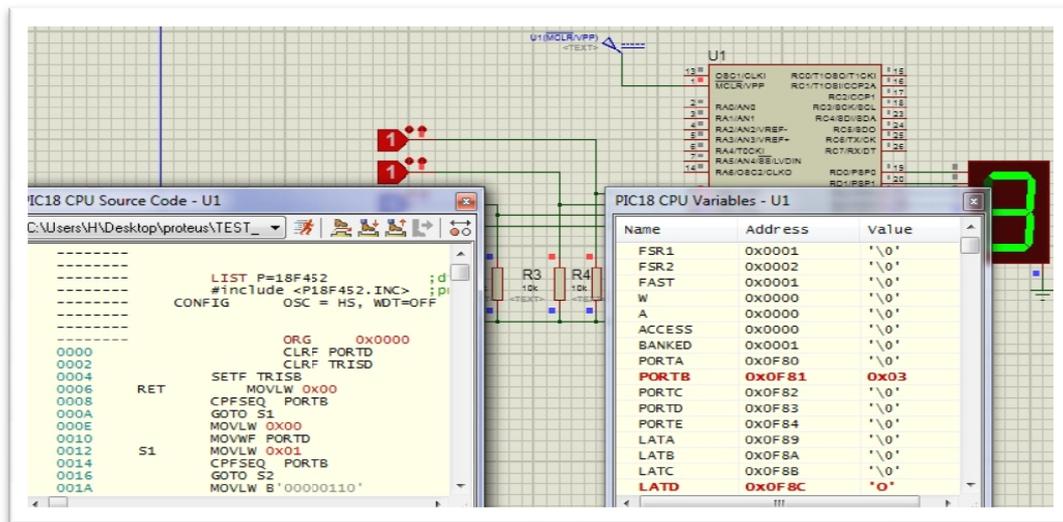
S1    MOVLW 0x01
      CPFSEQ PORTB
      GOTC S2
      MOVLW B'00000110'
      MOVWF PORTD

S2    MOVLW 0x03
      CPFSEQ PORTB
      GOTC RET
      MOVLW B'01001111'
      MOVWF PORTD
      GOTC RET
      END
    
```

6.2 Le mode pas à pas

La simulation en mode pas à pas sous PROTEUS est possible avec un affichage du programme associé au microcontrôleur ainsi que ses registres internes. Il est même possible d'ajouter des points d'arrêts afin de permettre un meilleur débogage.

Le fichier qui sera associé au microcontrôleur ne sera plus un exécutable de type (*.hex) mais plutôt de type (*.cof), vu qu'il contient le fichier source et les directives de désassemblage.



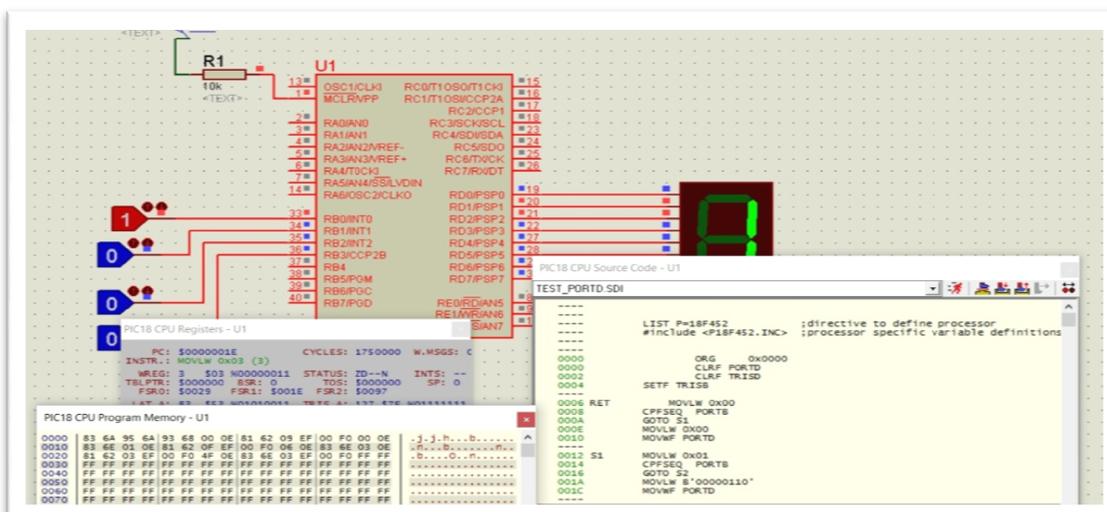
Application 3

- Exécuter l'application 2 en mode pas à pas.
- Modifier le programme proposé, pour permettre un décodage complet BCD/7Segments des 4 pins du PORTB ainsi qu'un affichage simultané sur le 7 segments

Remarque

Les versions de PROTEUS antérieures à la version 8, ne reconnaissent pas les fichiers de type (*.cof), mais il est tout de même possible de configurer le mode pas-à-pas par les étapes suivantes:

- Dans le menu « Source » Sélectionner 'Ajout/suppression fichiers source', puis cliquer sur 'New', et sélectionner le fichier source (*.asm).
- MPASM est l'outil de génération de code pour les microcontrôleurs PIC qu'il faut choisir. (ce compilateur est lancé par PROTEUS durant l'animation, et générera le code machine (*.hex))
- le fichier exécutable (*.hex) doit aussi être associé au microcontrôleur (le champ 'Program File ' du composant.
- Après le lancement du mode DEBUG, les fenêtres de mise au point peuvent être affichées par l'intermédiaire du menu 'Debug' (les fenêtres: 'Registers', 'Source Code', et 'Data Memory').



6.3 Rappel sur MPLAB

L'environnement de développement intégré **MPLAB** fournit par le constructeur **Microchip**, est un outil qui regroupe un éditeur de texte, un compilateur **MPASM**, un outil de simulation, et un logiciel de programmation. Par conséquent il y a une possibilité de l'associer à **PROTEUS** pour constituer une plateforme de développement et de simulation très utile.

Dans cette section on décrira en bref comment créer un projet sous **MPLAB**, puis comment l'associer à **PROTEUS** via un ensemble d'exemples.

La notion de projet

Un projet est un fichier (extension : *.mcp) qui permet de gérer entre autres les fichiers de code assembleur, la référence et les paramètres du microcontrôleur ainsi que les options et les outils de débogage.

Par conséquent pour chaque application il est intéressant de définir un nouveau projet.

Création d'un nouveau projet :

Après le lancement de MPLAB, On peut créer un nouveau projet :

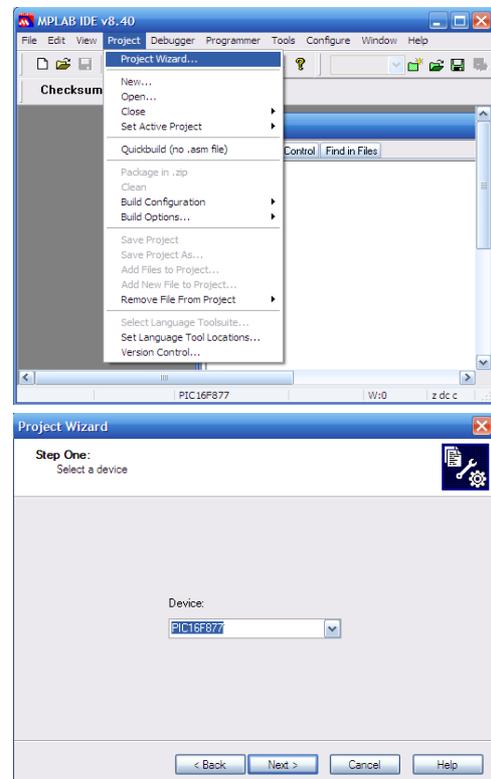
- Soit en passant par **project** → **New**,
Où on sera ramené à définir un nom et un répertoire pour le projet.
- Soit en passant par l'assistant 'Project Wizard' qui sera utilisé par la suite.

Définition du projet avec l'assistant

- Dans le menu **Project**, sélectionner **Project Wizard**. Cela lance un assistant permettant de définir certaines options du projet.

- La première étape consiste à Sélectionner d'abord un microcontrôleur

- L'étape suivante permet le choix des programmes de développement .on gardera l'ensemble logiciel de **Microchip**



- Ensuite, il faut choisir un nom et un emplacement pour le nouveau projet (utiliser **browse** pour spécifier le nom et le répertoire de travail).

- La fenêtre suivante permet l'ajout de fichiers existants au projet (cette phase n'est pas toujours nécessaire).

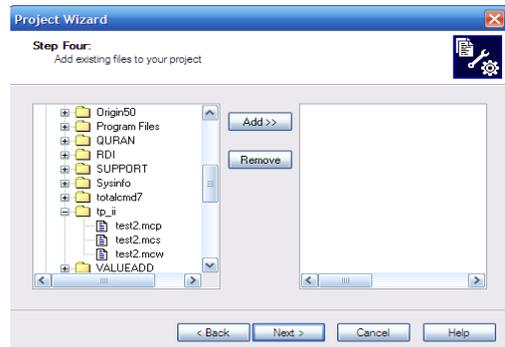
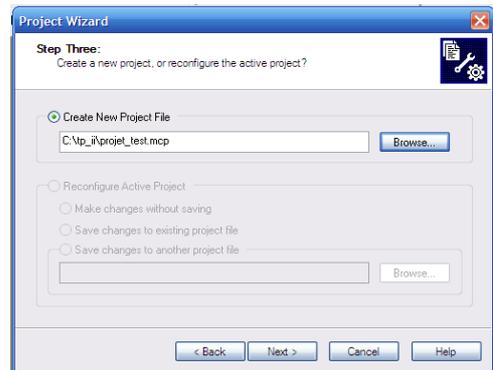
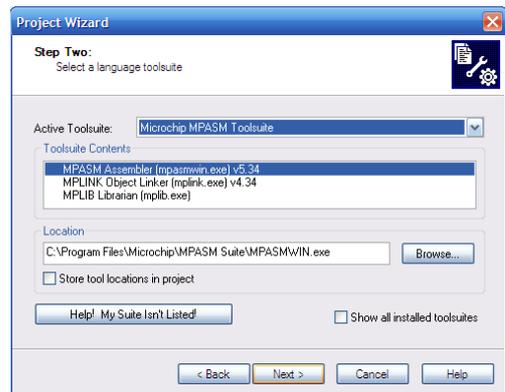
- La dernière fenêtre du **wizard** résume les principaux paramètres du projet, qui est automatiquement créée après avoir cliqué sur '**finish**'.

Remarque

- Le projet ainsi créé est vide, par conséquent il est nécessaire d'ajouter en particulier des fichiers sources (en assembleur).

- Une fois le fichier source (.asm) est édité utiliser (**project → add files to project**) pour l'ajouter au projet.

- Pour la sauvegarde du projet utiliser (**project → save project**)



Changement de compilateur

Sous l'environnement MPLAB, la création d'un projet peut se faire en utilisant d'autre compilateur que l'assembleur, en particulier les compilateurs C (par exemple CCS ou C18). Dans ce cas les étapes de configuration sont les suivantes :

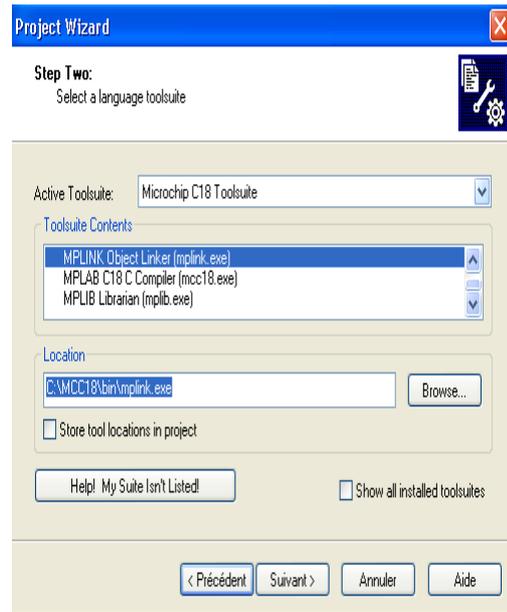
1. Lancez MPLAB
2. Créer un nouveau projet, et sélectionner un répertoire de sauvegarde.
3. Si vous utilisez le '**project wizard**' il faut s'assurer au niveau de l'étape 2 de la sélection d'un compilateur(exemple: **C18 "Microchip C18 Toolsuite"**)

En cas d'oubli réaliser cette configuration en passant par :

Project>Select Language Toolsuite

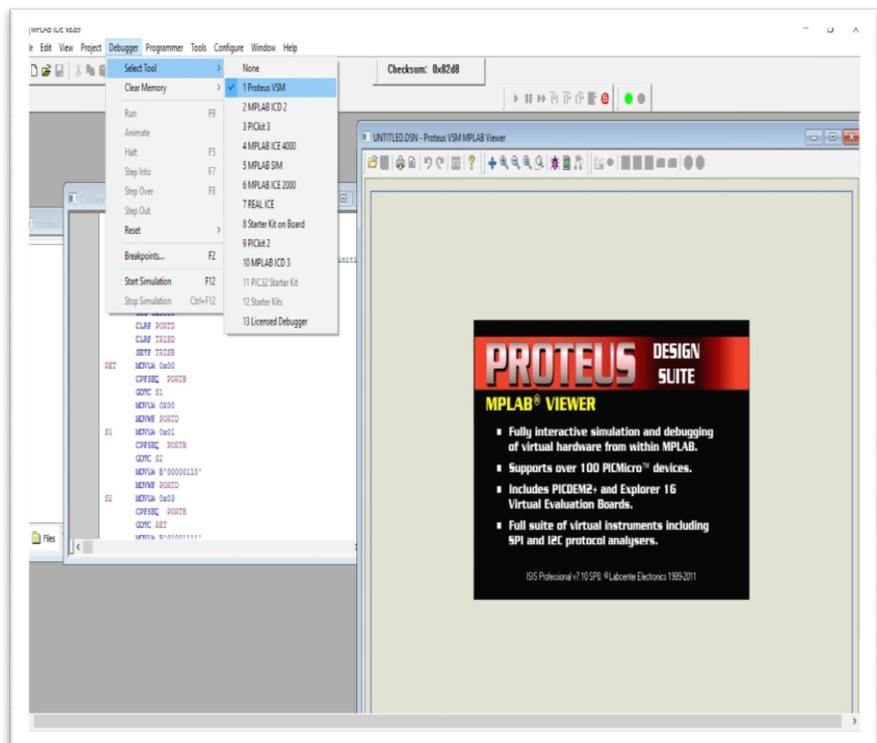
4. Cliquer sur le menu **project>Build option>Project** une fenêtre **Build option** s'ouvre, dans la sous fenêtre '**Directories and search paths**' cliquer sur '**Include search path**', vous cliquez sur **New** ensuite sur les points pour chercher le repertoire(exemple: **C:\MCC18\h**).

faites la même chose pour '**Library search path**'(le répertoire **C:\MCC18\lib**) ensuite la même chose pour '**Linker search path**'(le répertoire **C:\MCC18\lkr**)



6.4 L'Association de PROTEUS à MPLAB

Le simulateur PROTEUS est intégré à l'environnement de développement MPLAB comme outil de débogage, chose qui facilite la réalisation rapide d'applications. Après le lancement de MPLAB, et la création d'un nouveau projet il suffit par l'intermédiaire du menu 'Debug' (Select tool>> Proteus VMS) d'ouvrir une nouvelle fenetre pour afficher le schema réalisé par proteus.



Le module ARES de conception des circuits imprimés

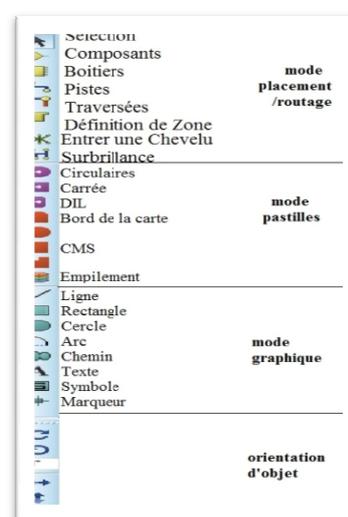
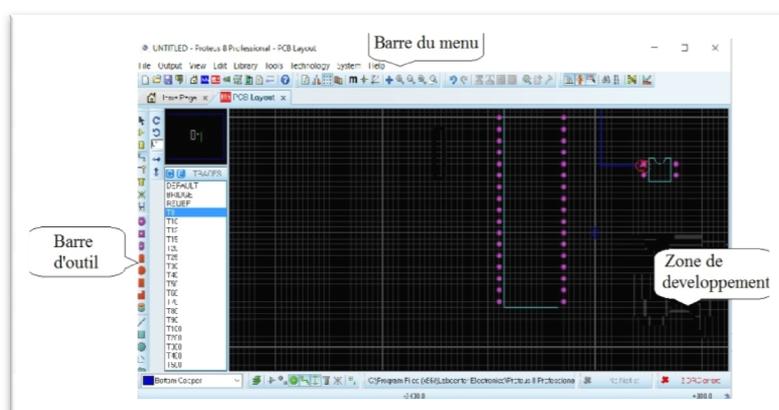
Dans cette partie, on s'intéresse au logiciel ARES, qui est un module de ISIS-PROTEUS permettant la conception de circuits imprimés.

Ce logiciel permet un placement de composants, ainsi qu'un routage sur 16 couches, soit en mode manuel, semi-automatique, ou automatique.

les algorithmes implémentés dans ce logiciel permettent une optimisation de la conception, avec même des outils supplémentaires dans les dernières versions tel que la visualisation 3D de la carte réalisée.

7.1 Présentation de l'interface ARES

Le lancement d' ARES donne accès à aune interface de développement similaire à celle de PROTEUS.

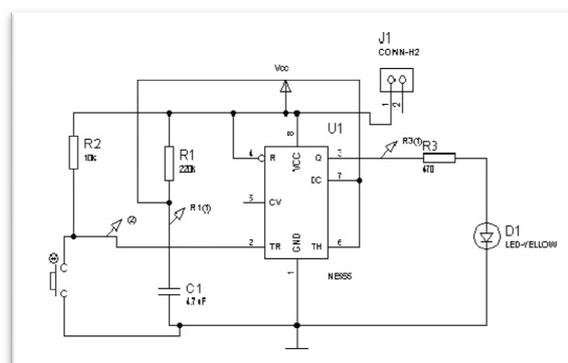


La barre d'outil offre quatre modes, dont les noms sont donnés dans la figure ci-contre. les principaux raccourcis seront définis et détaillés au fur et à mesure dans l'exemple suivant de routage automatique.

7.2 Routage automatique:

Il est possible de passer d'un montage simuler avec PROTEUS directement a la conception du circuit imprimé tel qu'il est expliquer dans l'exemple en cours.

On considère le montage astable ci-contre déjà étudié et simulé avec PROTEUS.

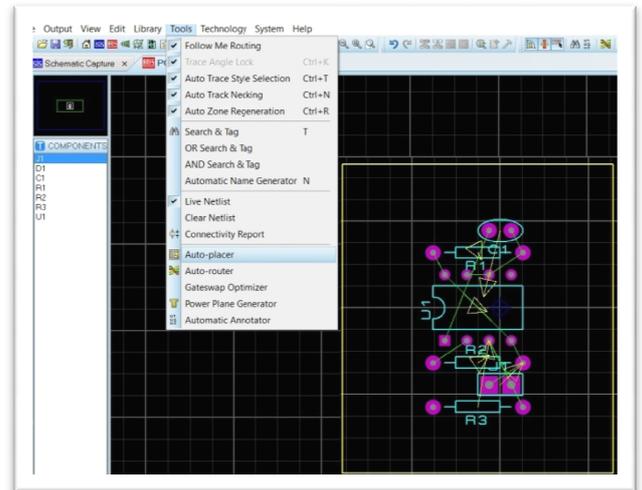


Après le lancement d'ARES, les étapes de base permettant la conception du circuit imprimé sont les suivantes:

* La première étape, consiste à utiliser le mode graphique (rectangle), pour le tracé et la délimitation des bords du circuit imprimé.

Durant ce tracé la couche 'Board Edge' doit être activée

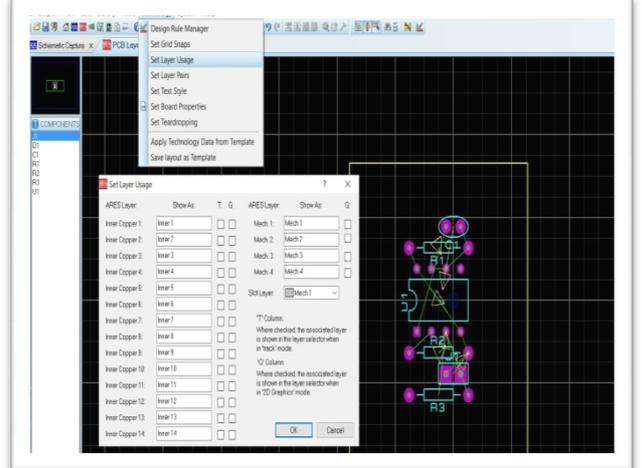
* La deuxième étape, consiste à placer les composants du circuit, qui peut être effectué automatiquement via "Tools>Auto-placer"



* Initialement, désactiver l'affichage des couches (non utilisées) dans "la liste des couches", afin de faciliter l'opération de conception du circuit imprimé.

remarques

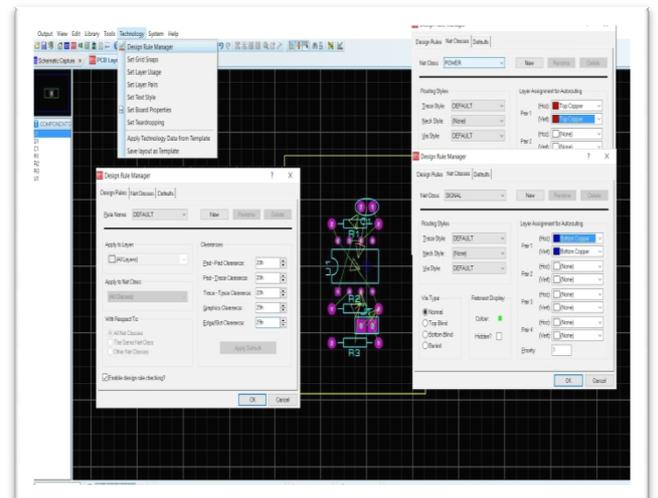
- La visibilité et la couleur des différentes couches est contrôlable via "View>Edit layer colour and Visibility".
- L'affichage de la grille, et le changement de l'unité de mesure des distances se font respectivement via "View>Toggle Grid" et "View>Toggle Metric/Imperial".



* Les règles de conception du circuit imprimé sont fixées via "Technology>Design Rule Manager". Elles définissent le type et taille des différentes pistes du circuit.

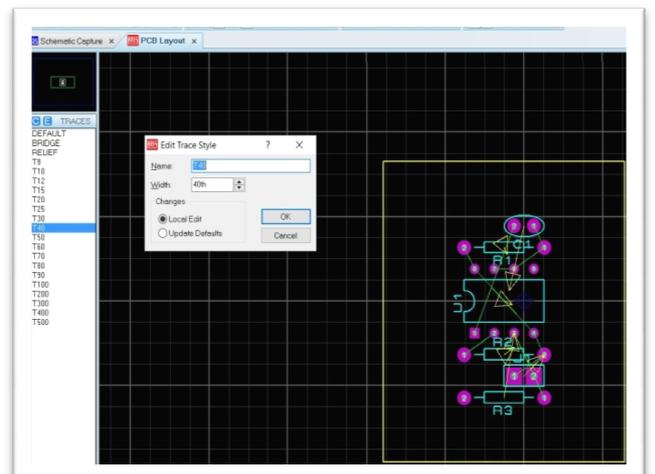
Ces règles de routage (sa stratégie de conception) sont définies pratiquement dès la saisie du schéma, et sont classées en trois différents types de routage :

- POWER : pour les signaux VCC et GND
- BUS : pour les liens nommés D[0]
- SIGNAL : pour le reste de liens



Un exemple est donné dans la figure ci-contre

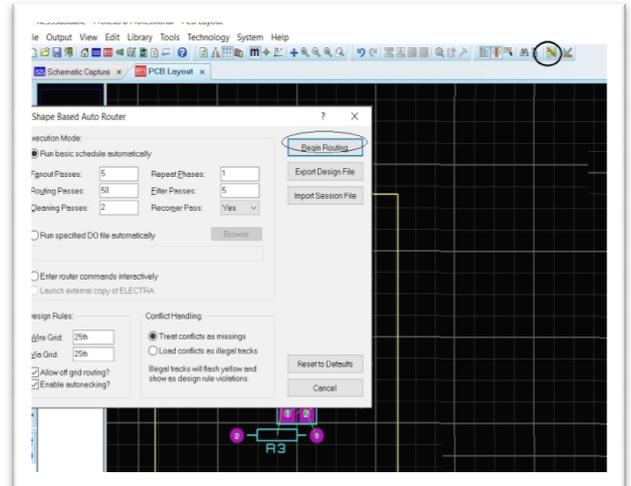
* Le mode "piste" (Trace) permet de définir la taille des pistes à concevoir durant le processus de routage.



* Le lancement du routage automatique est effectué via l'icone "auto-router (figure ci-contre)

Application 1:

refaire les étapes déjà décrites pour la conception d'un circuit double face, puis simple face du schéma proposé.

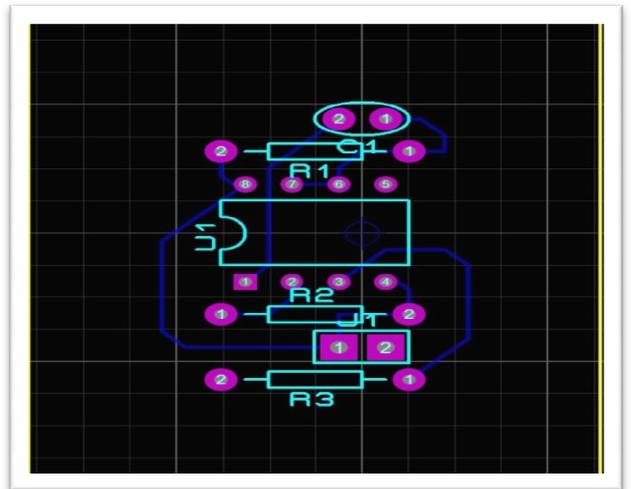


Application 2:

réaliser un circuit simple face du montage du montage astable (à base du NE555) déjà étudié.

Application 3:

réaliser un circuit double face du dernier montage à base du 18f452 déjà décrit et simulé dans la partie précédente.



Annexe

Dans cet annexe sont donnés des extraits des datasheets fournis par les constructeurs des composants suivants :

- L'amplificateur opérationnel TL082
- Le circuit multifonction NE555
- Le microcontrôleur PIC18F452

L'amplificateur opérationnel TL082

TL082 Wide Bandwidth Dual JFET Input Operational Amplifier

Check for Samples: TL082-N

FEATURES

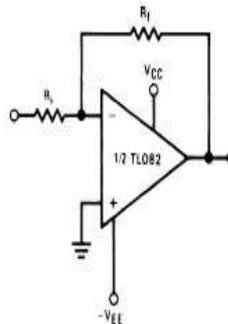
- Internally Trimmed Offset Voltage: 15 mV
- Low Input Bias Current: 50 pA
- Low Input Noise Voltage: 16nV/√Hz
- Low Input Noise Current: 0.01 pA/√Hz
- Wide Gain Bandwidth: 4 MHz
- High Slew Rate: 13 V/μs
- Low Supply Current: 3.6 mA
- High Input Impedance: $10^{12}\Omega$
- Low Total Harmonic Distortion: $\leq 0.02\%$
- Low 1/f Noise Corner: 50 Hz
- Fast Settling Time to 0.01%: 2 μs

DESCRIPTION

These devices are low cost, high speed, dual JFET input operational amplifiers with an internally trimmed input offset voltage (BI-FET II™ technology). They require low supply current yet maintain a large gain bandwidth product and fast slew rate. In addition, well matched high voltage JFET input devices provide very low input bias and offset currents. The TL082 is pin compatible with the standard LM1558 allowing designers to immediately upgrade the overall performance of existing LM1558 and most LM358 designs.

These amplifiers may be used in applications such as high speed integrators, fast D/A converters, sample and hold circuits and many other circuits requiring low input offset voltage, low input bias current, high input impedance, high slew rate and wide bandwidth. The devices also exhibit low noise and offset voltage drift.

Typical Connection



Connection Diagram

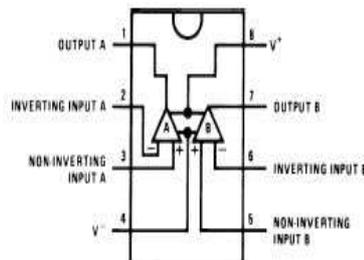
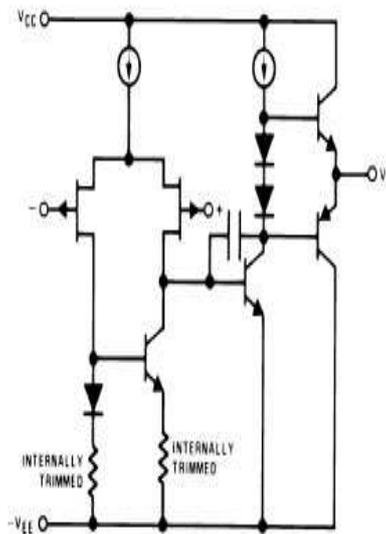


Figure 1. PDIP/SOIC Package (Top View)
See Package Number D0008A or P0008E

Simplified Schematic



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

Absolute Maximum Ratings ⁽¹⁾⁽²⁾

Supply Voltage	±18V
Power Dissipation ⁽³⁾	⁽⁴⁾
Operating Temperature Range	0°C to +70°C
T _j (MAX)	150°C
Differential Input Voltage	±30V
Input Voltage Range ⁽⁵⁾	±15V
Output Short Circuit Duration	Continuous
Storage Temperature Range	-65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	260°C
ESD rating to be determined.	

- (1) "Absolute Maximum Ratings" indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not ensure specific performance limits.
- (2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
- (3) The power dissipation limit, however, cannot be exceeded.
- (4) For operating at elevated temperature, the device must be derated based on a thermal resistance of 115°C/W junction to ambient for the P0008E package.
- (5) Unless otherwise specified the absolute maximum negative input voltage is equal to the negative power supply voltage.

Le circuit multifonction NE555



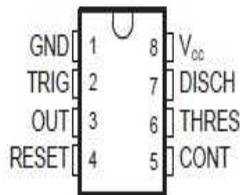
NA555, NE555, SA555, SE555

www.ti.com

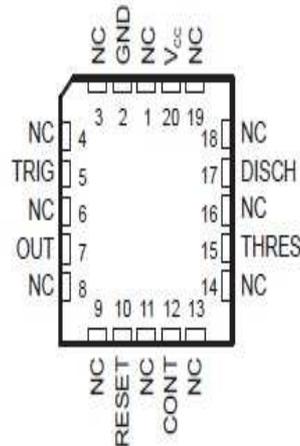
SLFS022I – SEPTEMBER 1973 – REVISED SEPTEMBER 2014

6 Pin Configuration and Functions

NA555...D OR P PACKAGE
NE555...D, P, PS, OR PW PACKAGE
SA555...D OR P PACKAGE
SE555...D, JG, OR P PACKAGE
(TOP VIEW)



SE555...FK PACKAGE
(TOP VIEW)



NC – No internal connection

Pin Functions

NAME	PIN		I/O	DESCRIPTION
	D, P, PS, PW, JG	FK		
	NO.			
CONT	5	12	I/O	Controls comparator thresholds, Outputs 2/3 VCC, allows bypass capacitor connection
DISCH	7	17	O	Open collector output to discharge timing capacitor
GND	1	2	–	Ground
NC		1, 3, 4, 6, 8, 9, 11, 13, 14, 16, 18, 19	–	No internal connection
OUT	3	7	O	High current timer output signal
RESET	4	10	I	Active low reset input forces output and discharge low.
THRES	6	15	I	End of timing input. THRES > CONT sets output low and discharge low
TRIG	2	5	I	Start of timing input. TRIG < 1/2 CONT sets output high and discharge open
Vcc	8	20	–	Input supply voltage, 4.5 V to 16 V. (SE555 maximum is 18 V)

7 Specifications

7.1 Absolute Maximum Ratings⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V _{CC}	Supply voltage ⁽²⁾		18	V
V _I	Input voltage	CONT, RESET, THRES, TRIG		V _{CC} V
I _O	Output current		±225	mA
θ _{JA}	Package thermal impedance ⁽³⁾⁽⁴⁾	D package	97	°C/W
		P package	85	
		PS package	95	
		PW package	149	
θ _{JC}	Package thermal impedance ⁽⁵⁾⁽⁶⁾	FK package	5.61	°C/W
		JG package	14.5	
T _J	Operating virtual junction temperature		150	°C
	Case temperature for 60 s	FK package	260	°C
	Lead temperature 1.6 mm (1/16 in) from case for 60 s	JG package	300	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values are with respect to GND.
- (3) Maximum power dissipation is a function of T_{J(max)}, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is P_D = (T_{J(max)} - T_A) / θ_{JA}. Operating at the absolute maximum T_J of 150°C can affect reliability.
- (4) The package thermal impedance is calculated in accordance with JESD 51-7.
- (5) Maximum power dissipation is a function of T_{J(max)}, θ_{JC}, and T_C. The maximum allowable power dissipation at any allowable case temperature is P_D = (T_{J(max)} - T_C) / θ_{JC}. Operating at the absolute maximum T_J of 150°C can affect reliability.
- (6) The package thermal impedance is calculated in accordance with MIL-STD-883.

7.2 Handling Ratings

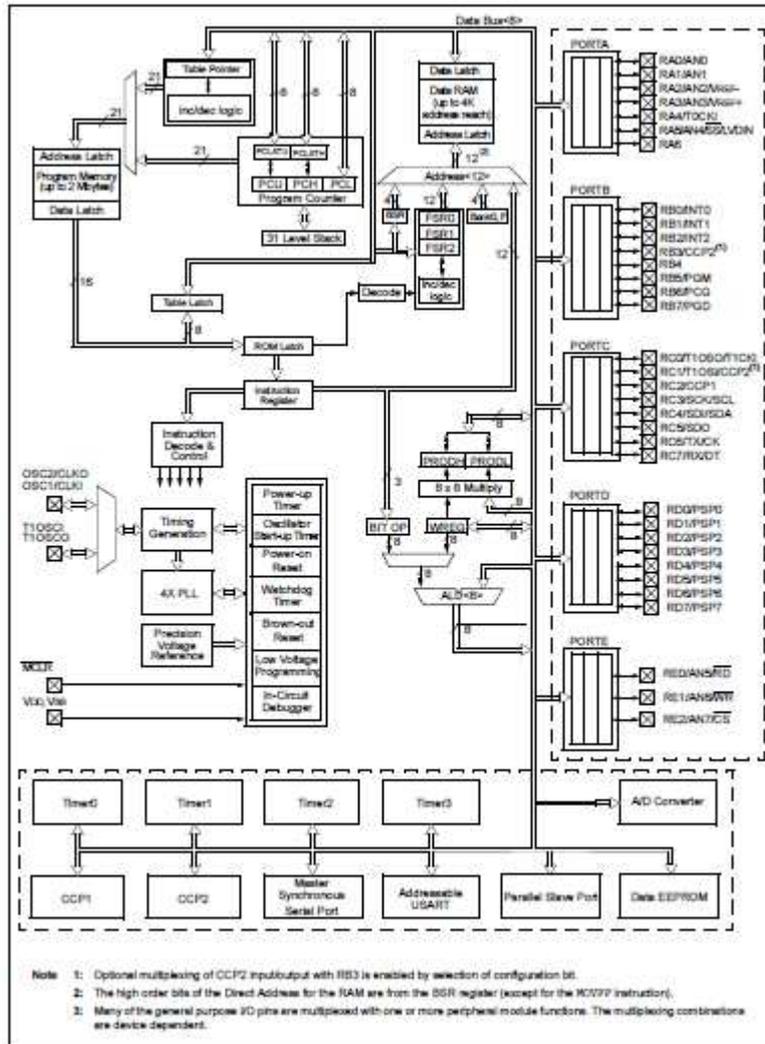
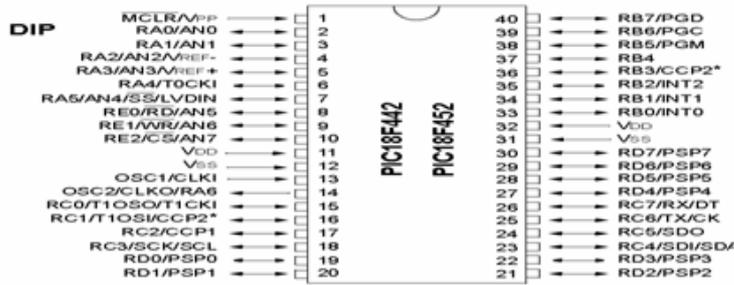
PARAMETER	DEFINITION	MIN	MAX	UNIT
T _{stg}	Storage temperature range	-65	150	°C

7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT	
V _{CC}	Supply voltage	NA555, NE555, SA555	4.5	16	V
		SE555	4.5	18	
V _I	Input voltage	CONT, RESET, THRES, and TRIG		V _{CC} V	
I _O	Output current		±200	mA	
T _A	Operating free-air temperature	NA555	-40	105	°C
		NE555	0	70	
		SA555	-40	85	
		SE555	-55	125	

Le microcontrôleur PIC18F452



PIC18FXX2

TABLE 20-2: PIC18FXXX INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word		Status Affected	Notes			
			MSb	LSb					
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d, a	Add WREG and f	1	0010	01da0	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	0010	0da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1, 2
CLRF	f, a	Clear f	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	4
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	4
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1, 2
DECF	f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1, 2
INCF	f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF	f, d, a	Move f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF	f _s , f _d	Move f _s (source) to 1st word f _d (destination) 2nd word	2	1100	ffff	ffff	ffff	None	
MOVWF	f, a	Move WREG to f	1	0110	111a	ffff	ffff	None	
MULWF	f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	None	
NEGF	f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	1, 2
RLCF	f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	1, 2
RRCF	f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	Set f	1	0110	100a	ffff	ffff	None	
SUBFWB	f, d, a	Subtract f from WREG with borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWF	f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	
SUBWFB	f, d, a	Subtract WREG from f with borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	1, 2
SWAPF	f, d, a	Swap nibbles in f	1	0011	10da	ffff	ffff	None	4
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, d, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- Note 2:** If this instruction is executed on the TMR0 register (and, where applicable, `d = 1`), the prescaler will be cleared if assigned.
- Note 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOB`.
- Note 4:** Some instructions are 2-word instructions. The second word of these instructions will be executed as a `NOB`, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
- Note 5:** If the Table Write starts the write cycle to internal memory, the write will continue until terminated.

TABLE 20-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb			LSb			
CONTROL OPERATIONS									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BNOV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	n	Branch if Not Zero	2	1110	0001	nnnn	nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	n	Branch Unconditionally	1 (2)	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	n, s	Call subroutine 1st word 2nd word	2	1110	110n	kkkk	kkkk	None	
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	\overline{TO} , \overline{PD}	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	C	
GOTO	n	Go to address 1st word 2nd word	2	1110	1111	kkkk	kkkk	None	
NOP	—	No Operation	1	0000	0000	0000	0000	None	
NOP	—	No Operation	1	1111	xxxx	xxxx	xxxx	None	4
POP	—	Pop top of return stack (TOS)	1	0000	0000	0000	0110	None	
PUSH	—	Push top of return stack (TOS)	1	0000	0000	0000	0101	None	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	None	
RESET	—	Software device RESET	1	0000	0000	1111	1111	All	
RETFIE	s	Return from interrupt enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	\overline{TO} , \overline{PD}	

- Note**
- 1: When a PORT register is modified as a function of itself (e.g., `MOVWF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
 - 2: If this instruction is executed on the TMR0 register (and, where applicable, $d = 1$), the prescaler will be cleared if assigned.
 - 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
 - 4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
 - 5: If the Table Write starts the write cycle to internal memory, the write will continue until terminated.

PIC18FXX2

TABLE 20-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb			LSb			
LITERAL OPERATIONS									
ADDLW	k	Add literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW	k	AND literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW	k	Inclusive OR literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR	f, k	Move literal (12-bit) 2nd word to FSRx 1st word	2	1110	1110	00ff	kkkk	None	
MOVLB	k	Move literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW	k	Move literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW	k	Multiply literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW	k	Subtract WREG from literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW	k	Exclusive OR literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS									
TBLRD*		Table Read	2	0000	0000	0000	1000	None	
TBLRD*+		Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-		Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD*+		Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*		Table Write	2 (5)	0000	0000	0000	1100	None	
TBLWT*+		Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*-		Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT*+		Table Write with pre-increment		0000	0000	0000	1111	None	

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- Note 2:** If this instruction is executed on the TMR0 register (and, where applicable, $d = 1$), the prescaler will be cleared if assigned.
- Note 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- Note 4:** Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
- Note 5:** If the Table Write starts the write cycle to internal memory, the write will continue until terminated.