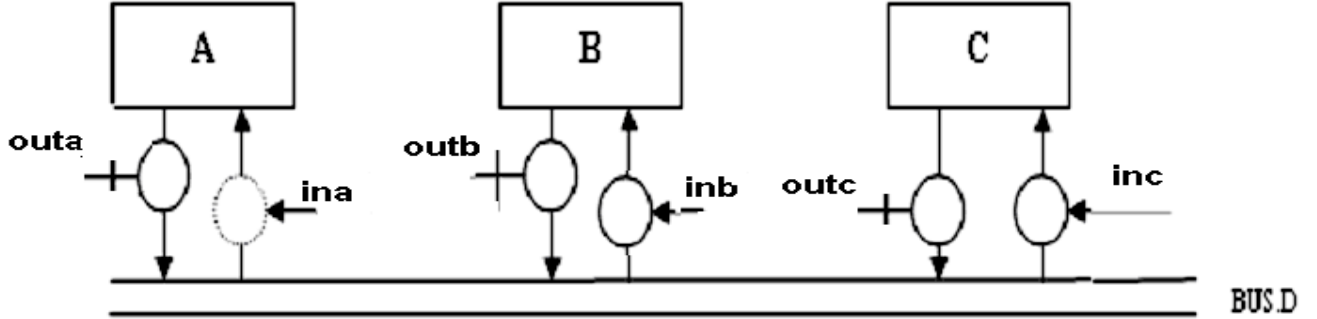


تصميم معالج صغري نموذجي للنقل ما بين ثلاثة مسجلات

يمكن للمعالجات الصغرية أن تنفذ أعداداً كبيرة جداً من التعليمات ، ولكن هنا سنتعرف على كيفية تصميم معالج صغري نموذجي ذي ثمان تعليمات فقط للنقل ما بين ثلاثة مسجلات فقط والتي لها ست نبضات سماحية (ثلاث قصيرة وثلاث طويلة) كما يبين الشكل (1-5).



الشكل (1-5) نبضات السماحية للنقل تلبين المسجلات A و B و C

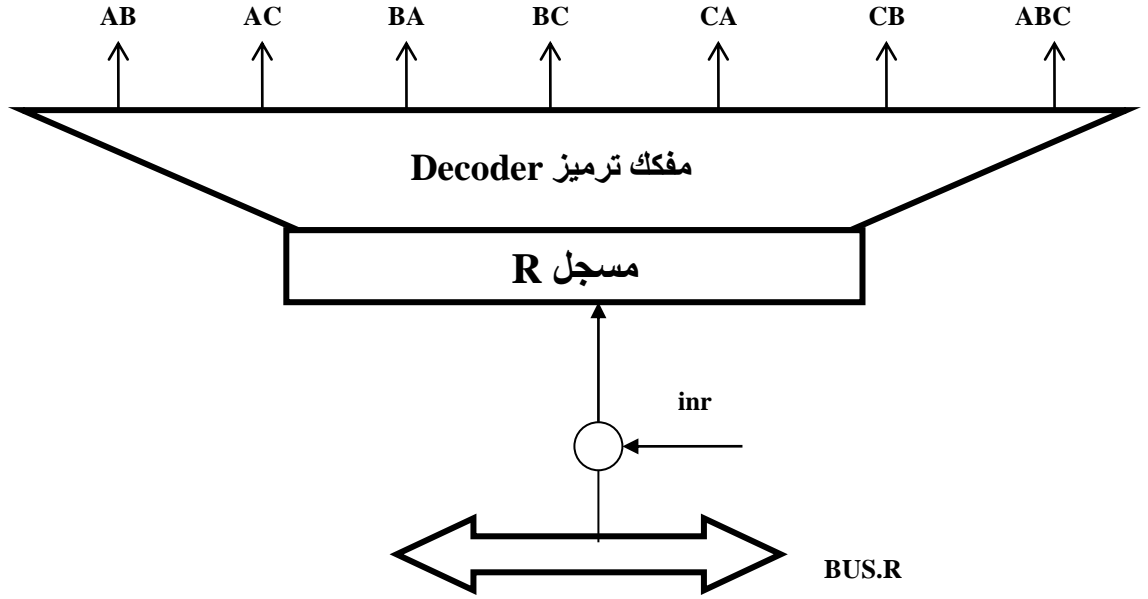
الجدول الآتي يبين ترميز Coding التعليمات مع تحديد تلك التعليمات بشكل حر.

المعنى	ترميز التعليمات	اسم التعليمات
(A) → B	001	AB نقل محتوى المسجل A إلى المسجل B
(A) → C	011	AC نقل محتوى المسجل A إلى المسجل C
(B) → A	100	BA نقل محتوى المسجل B إلى المسجل A
(B) → C	010	BC نقل محتوى المسجل B إلى المسجل C
(C) → A	101	CA نقل محتوى المسجل C إلى المسجل A
(C) → B	110	CB نقل محتوى المسجل C إلى المسجل B
(A) → B , C	111	ABC نقل محتوى المسجل A إلى المسجل B و إلى المسجل C
	000	لا تفعل شيء

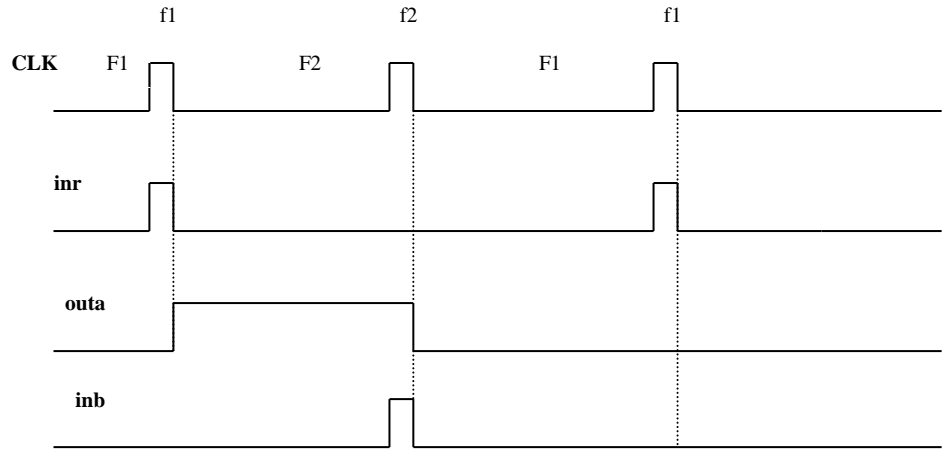
تحديد معاني التعليمات الثمان مع تحديد عمليات النقل الصغرية اللازمة لتنفيذها موضحة في الجدول أعلاه لندرس أولاً آلية فك ترميز التعليمات حيث من خلال مفكك ترميز decoder يتم تحديد التعليمات السبع المشار إليها وهي AB, AC, BA, BC, CA, CB هذا له ثلاثة مداخل و سبع مخرج ، المداخل قادمة من مسجل رابع، وليكن اسمه R وهو ذي ثلاث خانوات (حسب طول ترميز التعليمات). لنفرض أن ترميز التعليمات يأتي من ممرات خاصة غير الممرات BUS.D وليكن اسمها BUS.R . لاحظ الشكل (2-5).

نلاحظ بأنه أصبح لدينا سبع نبضات للسماحية بالنقل ما بين المسجلات الثلاثة وهي تطبق زمنياً على مرحلتين:

- مرحلة جلب التعليمات إلى المسجل R وفي هذه المرحلة تطبق النبضة inr فقط.
 - مرحلة تنفيذ التعليمات ، وفيها يتم تطبيق نبضتين واحدة قصيرة و الأخرى طويلة .
- والشكل (3-5) يبين المخطط الزمني لأول تعليمات فقط.



الشكل (2-5) دور مفك الترميز



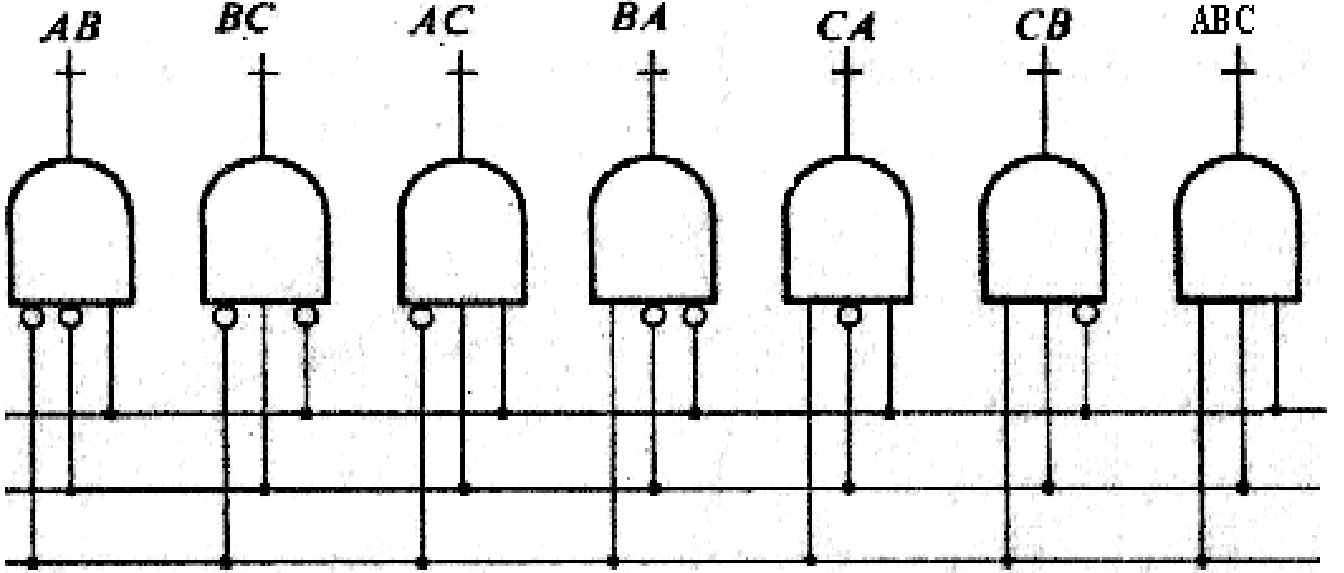
الشكل (3-5) المخطط الزمني للتعليلة AB فقط

لنرسم الآن دائرة مفك ترميز التعليمات على الشكل (4-5). النبضات f_1 و F_2 و f_2 تولد بمساعدة مولد نبضات بطورين وتزامن تنفيذ كل تعليمة من التعليمات السبع يكون على النحو الآتي:

$$AB = \begin{cases} inr = f_1 \\ outa = F_2 \\ inb = f_2 \end{cases}$$

$$AC = \begin{cases} inr = f_1 \\ outa = F_2 \\ inc = f_2 \end{cases}$$

$$BA = \begin{cases} inr = f_1 \\ outb = F_2 \\ ina = f_2 \end{cases}$$



الشكل (4-5) بنية مفكك الترميز

$$BC = \begin{cases} inr = f_1 \\ outb = F_2 \\ inc = f_2 \end{cases}$$

$$CA = \begin{cases} inr = f_1 \\ outc = F_2 \\ ina = f_2 \end{cases}$$

$$CB = \begin{cases} inr = f_1 \\ outc = F_2 \\ inb = f_2 \end{cases}$$

$$ABC = \begin{cases} inr = f_1 \\ outa = F_2 \\ inb, inc = f_2 \end{cases}$$

من خلال العلاقات السبع الأخيرة يمكن استنتاج الدوال (التوابع) المنطقية الآتية:

$$inr = f_1$$

$$outa = F_2(AB + AC + ABC)$$

$$outb = F_2(BA + BC)$$

$$outc = F_2(CA + CB)$$

$$ina = f_2(BA + CA)$$

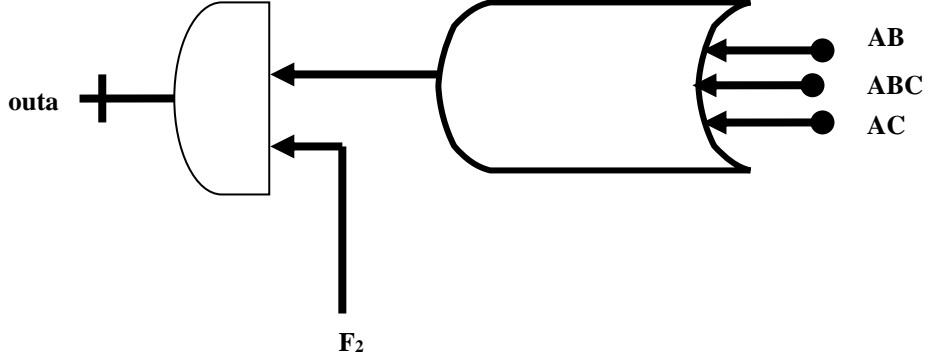
$$inb = f_2(AB + CB + ABC)$$

$$inc = f_2(AC + BC + ABC)$$

أي لدينا سبع دوال منطقية يمكننا تمثيلها بالدارات المنطقية (بوابات) و يمكننا أن نتمثيلها بالبرمجة الصغيرة Microprogramming والتي تعود فكرتها لـ Wilkes من عام 1951م حيث منذ ذلك الوقت أصبحت شائعة و استخدمت ضمن بنية المعالجات الصغيرة كذاكرات ثابتة ROM تحتوي على برامج صغيرة دائمة.

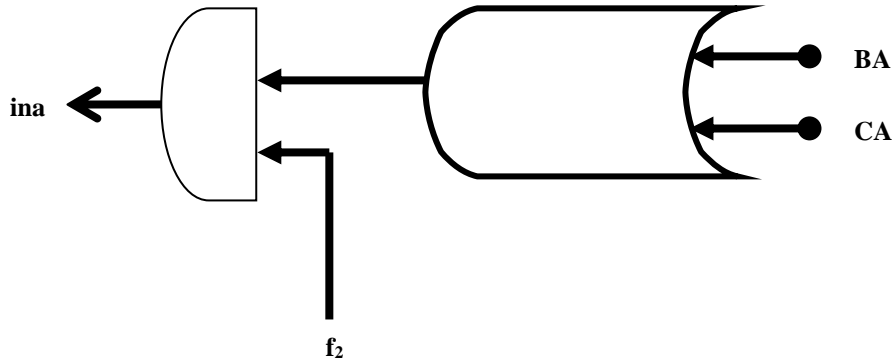
• التمثيل بالبوابات:

الشكل(5-5) يوضح آلية توليد النبضة outa



الشكل(5-5) دائرة توليد النبضة outa

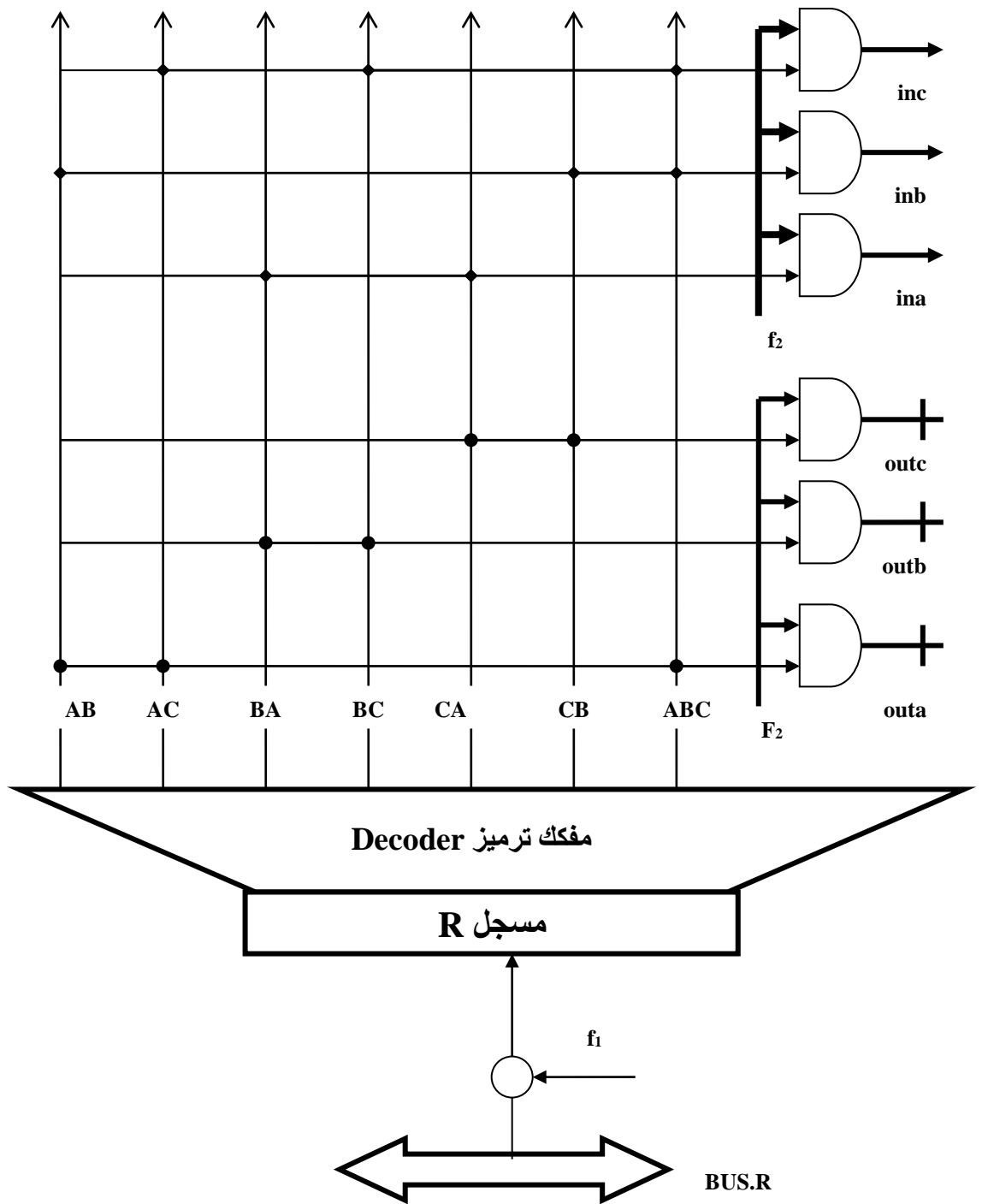
و الشكل(6-5) يوضح آلية توليد النبضة ina



الشكل(6-5) دائرة توليد النبضة ina

• التمثيل بالبرمجة الصغيرة:

الشكل(7-5) يوضح آلية توليد النبضات الست اللازمة.



الشكل (7-5) توليد النبضات بالبرمجة الصغيرة