

Pour ce système à six commutateurs, les sorties de l'additionneur parallèle sont connectées au décodeur DCB-sept segments qui pilote l'afficheur à sept segments. Tel que mentionné, il faut inclure des circuits supplémentaires pour augmenter les capacités du système.

Les résistances entre les entrées de chaque additionneur complet et la masse assurent un niveau BAS sur chaque entrée lorsque des commutateurs sont sur la position neutre (une logique CMOS est utilisée). Lorsqu'un commutateur est déplacé sur la position « oui » ou la position « non », un niveau HAUT (V_{CC}) est appliqué à l'entrée de l'additionneur complet correspondant.

SECTION 6-2
RÉVISION

1. Deux nombres de quatre bits, 1101 et 1011, sont appliqués à un additionneur parallèle de quatre bits. La retenue d'entrée vaut 1. Déterminez la somme (Σ) et la retenue de sortie.
2. Combien faut-il d'additionneurs 74LS283 pour faire la somme de deux nombres binaires, si chacun d'eux peut représenter des nombres décimaux jusqu'à $1\ 000_{10}$?

6-3 ADDITIONNEURS À PROPAGATION DE RETENUE ET À ANTICIPATION DE RETENUE

À la section précédente, nous avons vu qu'il existe deux catégories d'additionneurs parallèles, selon la manière dont ceux-ci traitent les retenues internes d'un étage à l'autre. Ces catégories sont les additionneurs à propagation de retenue et les additionneurs à anticipation de retenue. Quoiqu'ils soient similaires en termes d'entrée et de sortie, ces deux types d'additionneurs diffèrent au point de vue vitesse. Les additionneurs à anticipation de retenue sont beaucoup plus rapides que ceux à propagation de retenue.

Après l'étude de cette section, vous pourrez

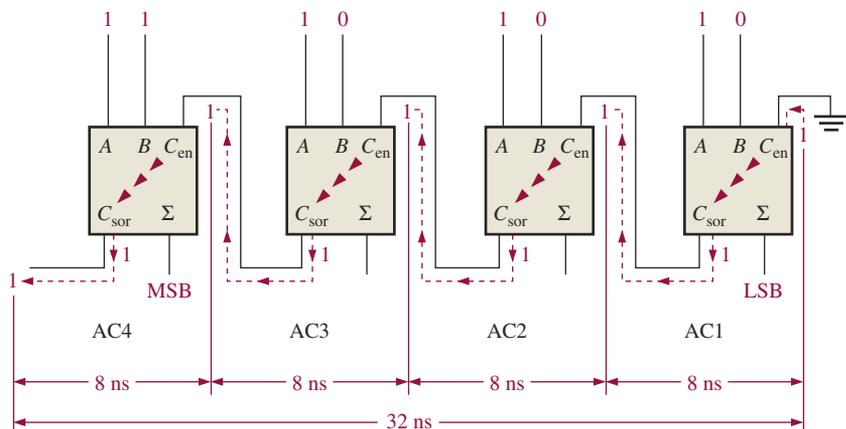
- Discuter des distinctions entre un additionneur à propagation de retenue et un additionneur à anticipation de retenue
- Énoncer l'avantage d'un additionneur à anticipation de retenue
- Définir et distinguer *génération de retenue* et *propagation de retenue*
- Développer une logique à anticipation de retenue
- Expliquer pourquoi un montage en cascade de CI 74LS283 exhibe à la fois des propriétés de propagation de retenue et d'anticipation de retenue

Additionneur à propagation de retenue

Dans un additionneur à **propagation de retenue**, la sortie de retenue de chaque additionneur complet est connectée à l'entrée de retenue de l'étage de rang plus élevé suivant (un étage désigne un additionneur complet). Comme la somme et la retenue de sortie d'un étage ne peuvent se produire que lorsqu'une retenue d'entrée est appliquée, cette méthode a pour effet de retarder le processus d'addition, comme l'illustre la figure 6-15. Le délai de propagation de retenue de chaque étage est égal à l'intervalle entre l'application de la retenue d'entrée et la production de la retenue de sortie, en présupmant que les entrées *A* et *B* sont déjà présentes.

► **FIGURE 6-15**

Additionneur parallèle de quatre bits à propagation de retenue et illustration des délais maximaux de propagation.



L'additionneur complet 1 (AC1) ne peut produire de retenue de sortie tant qu'il ne reçoit pas de retenue d'entrée. De même, l'additionneur complet 2 (AC2) ne peut générer de retenue de sortie tant que l'additionneur complet 1 (AC1) ne produit pas de retenue de sortie. Suivant la même logique, l'additionneur complet 3 (AC3) ne peut générer de retenue de sortie aussi longtemps qu'AC1 ne produit pas de retenue de sortie, suivie de celle d'AC2 et ainsi de suite. La figure 6–15 montre que la retenue d'entrée à l'étage du poids le plus faible (LSB) doit se propager à travers tous les autres étages d'additionneurs avant qu'une somme finale ne soit produite. Ce retard cumulatif traversant tous les étages représente le temps maximal avant d'obtenir la somme. Le délai total varie selon le bit de retenue produit par chaque additionneur complet. Si l'addition de deux nombres ne produit aucune retenue (0) entre les différents étages, alors le temps requis pour obtenir l'addition est égal au délai de propagation d'un seul additionneur complet, soit entre l'application des bits de données aux entrées et l'occurrence d'une somme à la sortie.

Additionneur à anticipation de retenue

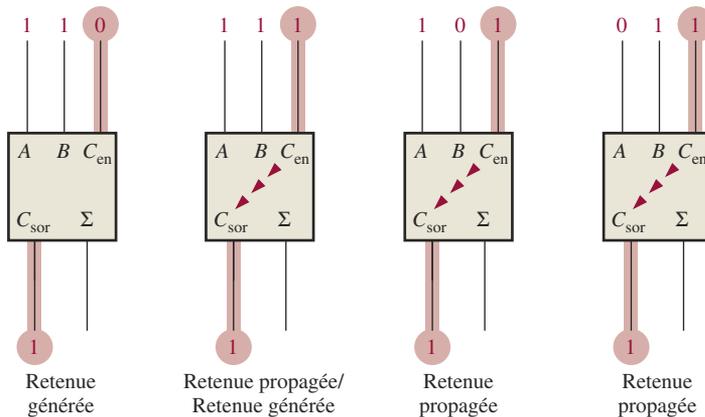
La vitesse avec laquelle une addition s'exécute dépend du temps requis par les retenues pour se propager à travers les étages d'un additionneur parallèle. L'une des méthodes pour accélérer le processus d'addition en éliminant ce délai de propagation est l'addition à **anticipation de retenue**. En appliquant cette méthode, il est possible d'anticiper la retenue de sortie de chaque étage et de produire, à partir des entrées, la retenue de sortie par génération ou par propagation.

Une **génération de retenue** se produit lorsqu'une retenue de sortie interne est générée par l'additionneur complet. Une retenue ne peut avoir lieu que lorsque les deux bits d'entrée sont des 1. La retenue générée, C_g , équivaut à la mise en opération ET des bits d'entrée A et B .

$$C_g = AB$$

Une **propagation de retenue** est créée lorsqu'une retenue d'entrée est répercutée pour devenir une retenue de sortie. Dans un additionneur complet, la propagation d'une retenue d'entrée peut avoir lieu lorsqu'au moins un bit d'entrée vaut 1. La retenue propagée, C_p , équivaut à la mise en opération OU des bits d'entrée.

$$C_p = A + B$$



La figure 6–16 illustre les conditions requises pour une génération de retenue et une propagation de retenue. Les triples pointes de flèches symbolisent la répercussion (propagation) des bits d'entrée.

La retenue de sortie d'un additionneur complet peut s'exprimer en tant que retenue propagée (C_p) ou en tant que retenue générée (C_g). La retenue de sortie (C_{sor}) vaut 1 si la retenue générée vaut 1 OU si la retenue propagée vaut 1 ET que la retenue d'entrée (C_{en}) vaut 1. En d'autres termes, nous obtenons une retenue de sortie de 1 si elle est générée par l'additionneur complet ($A = 1$ ET $B = 1$) ou si l'additionneur propage la retenue d'entrée ($A = 1$ OU $B = 1$) ET ($C_{en} = 1$). Cette relation s'exprime avec l'équation

$$C_{sor} = C_g + C_p C_{en}$$

Équation 6–5

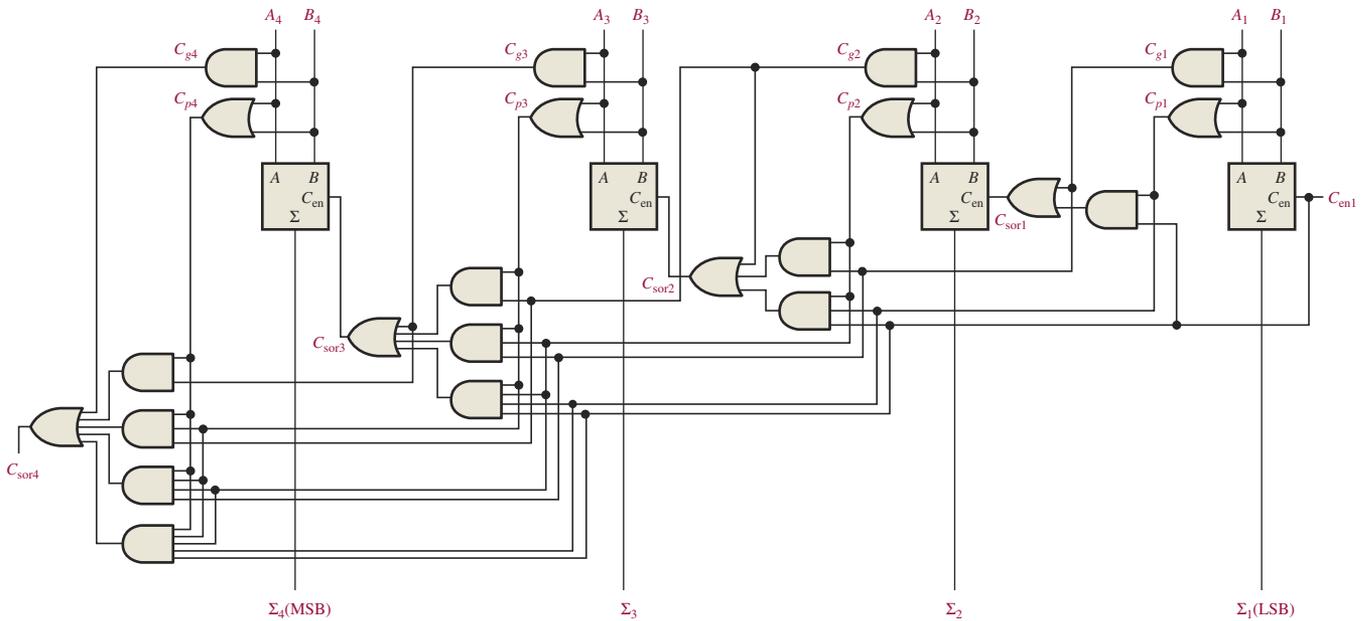
Équation 6–6

◀ FIGURE 6–16

Conditions pour une génération de retenue et une propagation de retenue.

Équation 6–7

Les équations C_{sor} sont mises en œuvre avec des portes logiques, puis reliées aux additionneurs complets pour créer un additionneur à anticipation de retenue de quatre bits, comme le montre la figure 6–18.



▲ FIGURE 6–18

Diagramme logique d'un additionneur à anticipation de retenue de quatre étages.

Combinaison d'additionneurs à anticipation et à propagation de retenue

Le CI 74LS283 présenté à la section 6–2 est un additionneur de quatre bits à anticipation de retenue. Lorsque de tels additionneurs sont mis en cascade pour traiter des nombres binaires de plus de quatre bits, il faut relier la sortie de retenue d'un circuit à l'entrée de retenue d'un autre, ce qui crée une condition de propagation entre les additionneurs. Par conséquent, l'additionneur résultant d'un tel montage combine l'anticipation et la propagation de retenue. Pendant que chaque additionneur MSI utilise à l'interne la méthode d'anticipation, la technique de propagation sert lorsqu'il faut répercuter la retenue de sortie d'un circuit vers l'entrée de retenue d'un autre additionneur.

SECTION 6–3 RÉVISION

1. Les bits d'entrée d'un additionneur complet sont $A = 1$ et $B = 0$. Déterminez C_g et C_p .
2. Déterminez la retenue de sortie d'un additionneur complet lorsque $C_{en} = 1$, $C_g = 0$ et $C_p = 1$.

6–4 COMPARETEURS

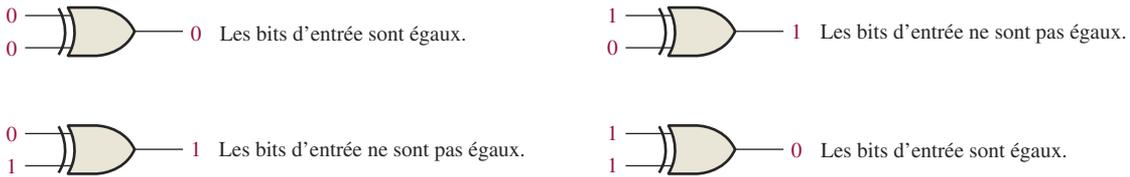
La fonction principale d'un comparateur est de comparer les grandeurs de deux quantités binaires afin de déterminer la relation existant entre ces quantités. Sous sa forme simplifiée, un circuit de comparateur détermine si deux nombres sont égaux.

Après l'étude de cette section, vous pourrez

- Utiliser une porte OU exclusif comme comparateur de base
- Analyser la logique interne d'un comparateur de grandeurs muni de sorties d'égalité et d'inégalité
- Utiliser un comparateur 74HC85 pour comparer deux nombres de quatre bits
- Faire un montage en cascade de CI 74HC85 pour créer un comparateur de huit bits ou plus

Égalité

Au chapitre 3, nous avons vu qu'une porte OU exclusif peut servir de comparateur de base puisque sa sortie vaut 1 si les deux bits à ses entrées diffèrent et qu'elle vaut 0 si les bits d'entrée sont identiques. La figure 6-19 illustre l'emploi d'une porte OU exclusif comme comparateur de deux bits.



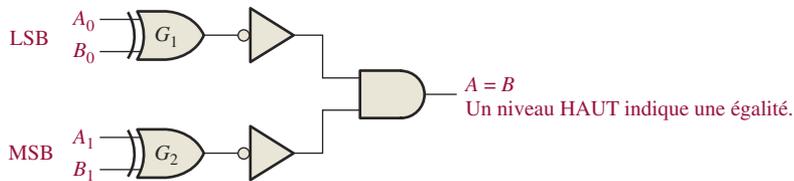
▲ FIGURE 6-19

Fonctionnement d'un comparateur de base.

Pour comparer des nombres binaires de deux bits chacun, il faut utiliser une porte OU exclusif additionnelle. Les bits les moins significatifs (LSB) des deux nombres sont comparés par la porte G_1 et les deux bits les plus significatifs (MSB) sont comparés par la porte G_2 , comme l'illustre la figure 6-20. Si les deux nombres sont égaux, leurs bits correspondants sont identiques et la sortie de chaque porte OU exclusif donne 0. Si les bits d'un même rang sont différents, un 1 est produit à la sortie de cette porte OU exclusif.

► FIGURE 6-20

Diagramme logique d'un comparateur d'égalité pour deux nombres de deux bits. Ouvrez le fichier F06-20 pour en vérifier le fonctionnement.



Forme générale: Nombre binaire $A \rightarrow A_1A_0$
 Nombre binaire $B \rightarrow B_1B_0$

Un comparateur détermine si deux nombres binaires sont égaux ou non.

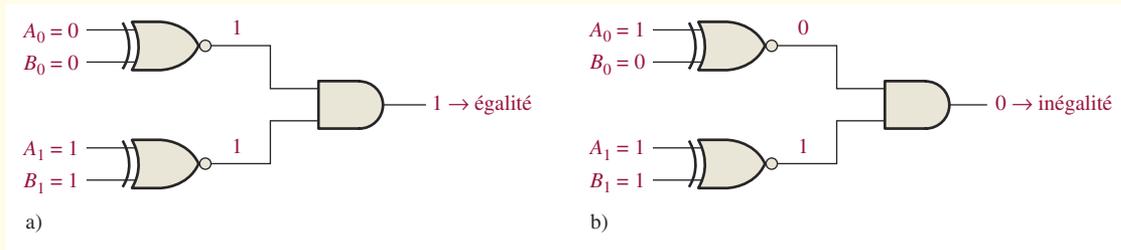
Pour générer une sortie simple indiquant une condition d'égalité ou d'inégalité entre deux nombres, on peut utiliser deux inverseurs et une porte ET, tel qu'illustré à la figure 6-20. La sortie de chaque porte OU exclusif est inversée et appliquée à l'entrée de la porte ET. Lorsque les deux bits d'entrée de chaque porte OU exclusif sont égaux, les bits correspondants des nombres sont égaux. Des 1 sont alors produits aux deux entrées de la porte ET, qui donne une sortie de 1. Lorsque les deux nombres ne sont pas égaux, au moins une série de bits inégaux d'un même rang produit un 0 sur une entrée de la porte ET, qui donne une sortie de 0. Par conséquent, la sortie de la porte ET indique l'égalité (1) ou l'inégalité (0) des deux nombres.

L'exemple 6-5 illustre ce fonctionnement avec deux cas spécifiques. La porte OU exclusif et l'inverseur sont remplacés par un symbole NON-OU exclusif.

EXEMPLE 6-5

Appliquez les groupes de nombres binaires suivants aux entrées du comparateur de la figure 6-21 et déterminez la sortie en suivant les niveaux logiques dans le circuit.

- a) 10 et 10
- b) 11 et 10



▲ FIGURE 6-21

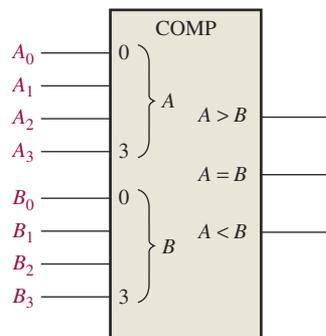
Solution a) Pour les entrées 10 et 10, la sortie donne **1**, comme l’illustre la figure 6–21 a).
 b) Pour les entrées 11 et 10, la sortie donne **0**, comme l’illustre la figure 6–21 b).

Exercice connexe Répétez ce processus pour les entrées binaires 01 et 10.

Au chapitre 3, vous avez vu qu’un comparateur de base peut être modifié pour accepter un grand nombre de bits. La porte ET établit la condition d’égalité pour tous les bits correspondants des deux nombres, si ceux-ci sont égaux.

Inégalité

En plus de la sortie d’égalité, la plupart des comparateurs à CI sont munis de sorties supplémentaires indiquant quel nombre binaire est le plus grand. Une sortie indique la condition lorsque le nombre *A* est plus grand que le nombre *B* ($A > B$) et une autre indique si le nombre *A* est plus petit que le nombre *B* ($A < B$), comme le montre le symbole logique du comparateur de quatre bits illustré à la figure 6–22.



◀ FIGURE 6-22

Symbole logique d’un comparateur de quatre bits avec des indications d’inégalité.



NOTE INFORMATIQUE

La *cache* d’un ordinateur est la mémoire intermédiaire très rapide entre l’unité centrale de traitement (UCT) et la mémoire principale plus lente. L’UCT sollicite les données en émettant leur *adresse* en mémoire (emplacement unique). Une partie de cette adresse contient un *préfixe*. Le *comparateur de préfixes des adresses* compare le préfixe de l’UCT et celui du répertoire de la cache. Si ces préfixes concordent, les données adressées sont déjà dans la cache et sont extraites très rapidement. Si les préfixes diffèrent, les données doivent être extraites de la mémoire principale à une vitesse beaucoup plus lente.

Pour déterminer une inégalité entre des nombres binaires *A* et *B*, vous devez d’abord examiner le bit le plus significatif de chaque nombre. Les conditions possibles sont les suivantes :

1. Si $A_3 = 1$ et $B_3 = 0$, le nombre *A* est plus grand que le nombre *B*.
2. Si $A_3 = 0$ et $B_3 = 1$, le nombre *A* est plus petit que le nombre *B*.
3. Si $A_3 = B_3$, vous devez examiner le rang de poids plus faible suivant pour identifier une condition d’inégalité.

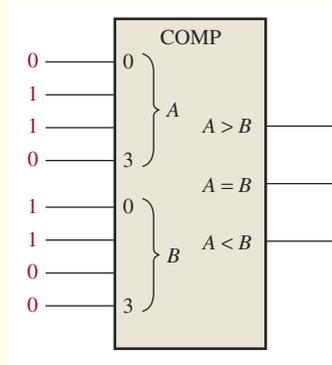
Ces trois opérations sont valides pour chaque rang de bit des nombres. La procédure générale utilisée par le comparateur est de vérifier une inégalité dans un rang de bits, en

débutant par les bits les plus significatifs (MSB). S'il y a inégalité, la relation entre les deux nombres est établie et toutes les autres inégalités possibles dans les autres rangs de bits de poids plus faibles sont ignorées, puisqu'une indication contraire pourrait se présenter. L'indication du rang le plus élevé doit toujours avoir préséance.

EXEMPLE 6-6

Déterminez les sorties $A = B$, $A > B$ et $A < B$ pour les nombres aux entrées du comparateur illustré à la figure 6-23.

► **FIGURE 6-23**



Solution Le nombre aux entrées A est 0110 et le nombre aux entrées B est 0011. **La sortie $A > B$ est au niveau HAUT et les autres sorties sont au niveau BAS.**

Exercice connexe Quelles sont les sorties du comparateur lorsque $A_3A_2A_1A_0 = 1001$ et que $B_3B_2B_1B_0 = 1010$?

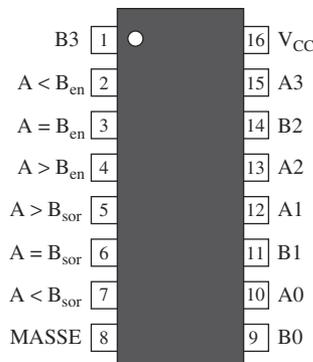
COMPARATEUR DE GRANDEURS DE QUATRE BITS 74HC85



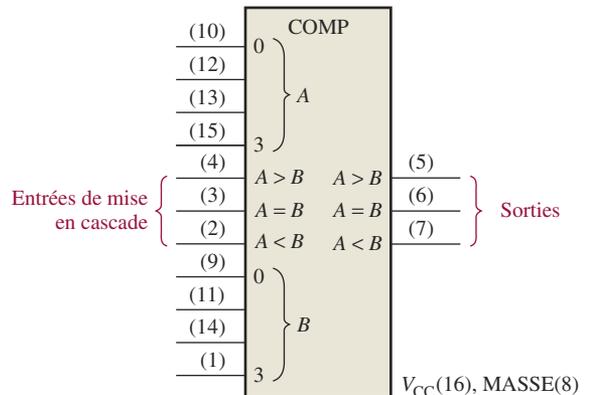
Le circuit 74HC85 est un comparateur disponible également dans d'autres familles de CI. La figure 6-24 illustre le diagramme des broches et le symbole logique. Ce composant contient non seulement toutes les entrées et sorties du comparateur discuté précédemment, mais comprend aussi trois entrées en cascade : $A < B$, $A = B$, $A > B$. Ces entrées permettent la mise en cascade de plusieurs comparateurs pour traiter des nombres de plus de quatre bits. Pour créer un comparateur plus puissant, les sorties $A < B$, $A = B$ et $A > B$ du comparateur

► **FIGURE 6-24**

Diagramme de broches et symbole logique du comparateur de grandeurs de quatre bits 74HC85. Le numérotage des broches est entre parenthèses.



a) Diagramme des broches



b) Symbole logique

des rangs moins significatifs sont connectées aux entrées en cascade correspondantes du comparateur des rangs plus élevés suivants. On doit aussi appliquer un niveau HAUT à l'entrée $A = B$ du comparateur des rangs plus faibles et des niveaux BAS sur ses entrées $A < B$ et $A > B$. Ce circuit est également disponible dans d'autres familles de type TTL et CMOS. Consultez le site Web de Texas Instruments à www.ti.com pour plus de détails.

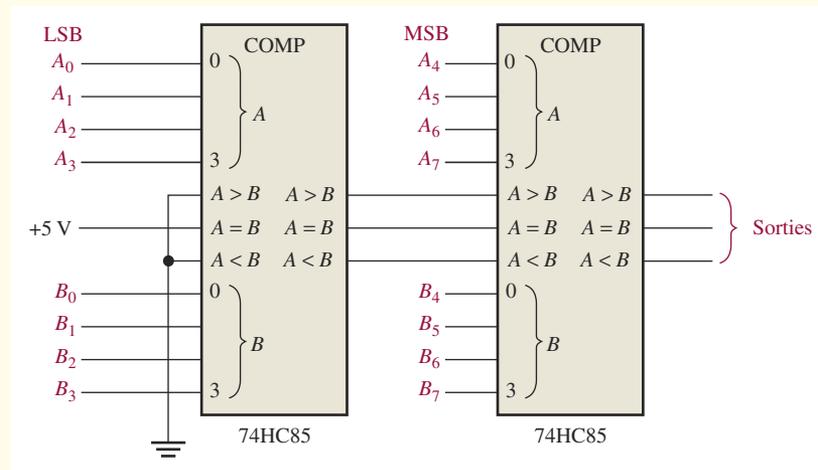
EXEMPLE 6-7

Utilisez des comparateurs 74HC85 pour comparer les grandeurs de deux nombres de huit bits. Illustrez les comparateurs et les interconnexions appropriées.

Solution Il faut deux composants 74HC85 pour comparer deux nombres de huit bits. Ils sont connectés dans un arrangement en cascade comme l'illustre la figure 6-25.

► **FIGURE 6-25**

Comparateur de grandeurs de huit bits utilisant deux circuits 74HC85.



Exercice connexe Développez le circuit de la figure 6-25 pour créer un comparateur de 16 bits.

SECTION 6-4 RÉVISION

1. Les nombres binaires $A = 1011$ et $B = 1010$ sont appliqués aux entrées d'un composant 74HC85. Déterminez les sorties.
2. Les nombres binaires $A = 11001011$ et $B = 11010100$ sont appliqués au comparateur de huit bits illustré à la figure 6-25. Déterminez les états aux broches de sortie 5, 6 et 7 de chaque composant 74HC85.

TRUC PRATIQUE

La plupart des composants CMOS sont munis d'un circuit de protection contre les décharges électrostatiques et les tensions électriques. Il faut toutefois prendre garde de ne jamais appliquer une tension supérieure aux tensions maximales recommandées. Pour un fonctionnement stable, les tensions d'entrée et de sortie doivent toujours être définies entre la masse et V_{CC} . En outre, n'oubliez pas de connecter les entrées inutilisées sur un niveau logique approprié (masse ou V_{CC}). Par contre, il n'est pas nécessaire de connecter les sorties inutilisées.